Algoritmos de visión para la estimación robusta de pose 3D

Autor: Marcos Iglesias García
Tutor: Nicolas Burrus
Cotutor: Mohamed Abderrahim

DEPARTAMENTO DE INGENIERÍA
DE SISTEMAS Y AUTOMÁTICA

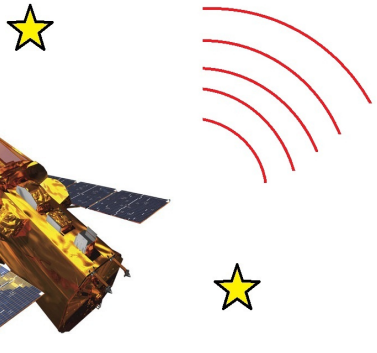
PROYECTO FIN DE CARRERA

UNIVERSIDAD CARLOS III DE MADRID

INGENIERÍA DE TELECOMUNICACIÓN

Leganés, octubre de 2010

dedicado a mis padres y a mi hermano Mario



Índice general

Índice de figuras

Índice de tablas

Parte I

Introducción

Capítulo 1

Introducción

El desarrollo de nuevos algoritmos en el campo de la visión artificial permite dotar a los robots móviles de una mayor autonomía en su funcionamiento. Si el autómata conoce el espacio que le rodea, es capaz de interactuar con el medio y realizar múltiples tareas. Tradicionalmente, los robots están equipados con microprocesadores, ordenadores a bordo que les permiten procesar la información que reciben y actuar en consecuencia. Sin embargo, carecen de inteligencia, lo que les impide reaccionar ante ambientes variables o escenarios desconocidos. Tareas como el reconocimiento, localización y manejo de objetos, suponen años de aprendizaje y experiencia para el ser humano. Implementar dichas propiedades en un ordenador supone un reto complicado y a pesar de que las capacidades en términos de velocidad de cómputo y memoria han incrementado de manera significativa en las últimas décadas, tan sólo se ha llegado a soluciones parciales.

Una de las tareas básicas que todos los robots móviles han de realizar es la de navegar y moverse en entornos desconocidos parcial o totalmente. En este sentido, han de identificar, en la medida de lo posible, el entorno en el que se encuentran así como los distintos objetos que interactúan en el. Autores como Levitt y Lawton [1] definen la navegación como el proceso que trata de responder las siguientes preguntas:

- ¿Dónde estoy?

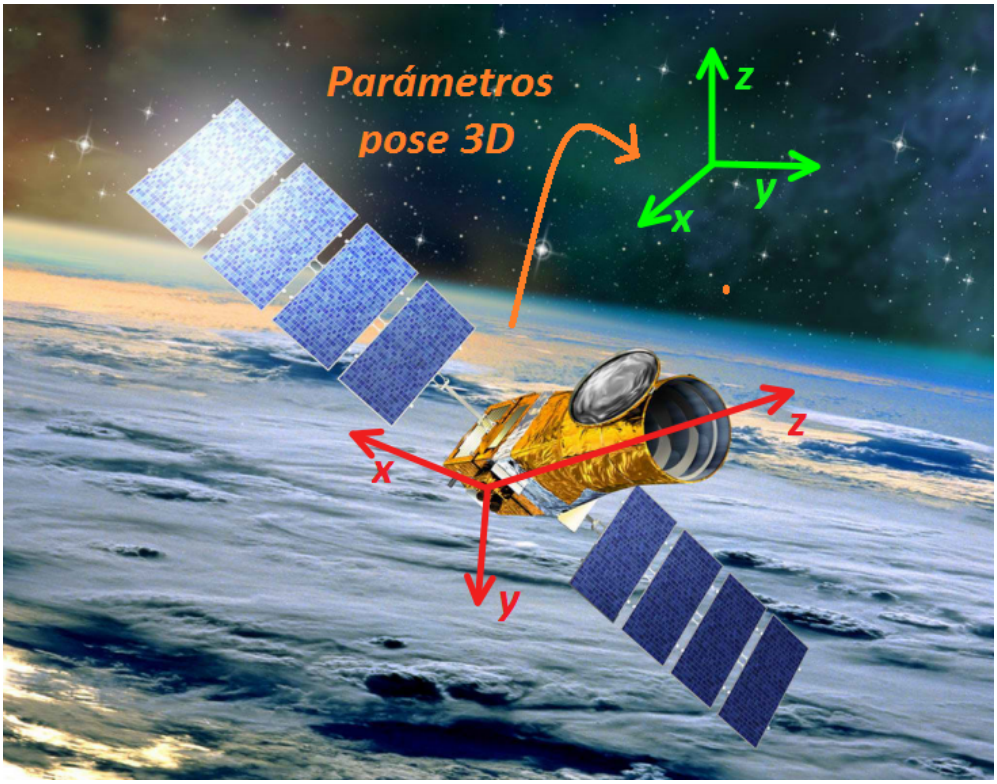


Figura 1.1: Pose 3D objeto. En la figura se presentan los sistemas de coordenadas del objeto bajo análisis (rojo) y el de la cámara (verde). La transformación entre ambos sistemas de coordenadas viene determinada por los parámetros de pose del objeto.

- ¿Dónde están otros objetos, lugares con respecto a mí?
- ¿Cómo puedo llegar a otros objetos, lugares desde mi posición?

En términos de posicionamiento y navegación, si conozco la posición de los objetos que me rodean, puedo determinar mi posición relativa y calcular trayectorias de aproximación hacia ellos. Además, tareas tales como la manipulación de dichos objetos son posibles, pudiendo responder a las preguntas formuladas con anterioridad.

El problema de estimación de ***pose 3D (rotación y translación)*** de un objeto en el espacio, se consolida como uno de los retos más destacados en el campo de la visión artificial. Como cabe esperar, el problema

genérico se desglosa en subproblemas tales como la detección y análisis del movimiento.

- El problema de *detección* tiene como objetivo determinar si cierto objeto aparece en la imagen. Esta tarea se resuelve de manera robusta y sin esfuerzo por el ser humano, sin embargo, todavía no está resuelto satisfactoriamente en el contexto de visión por ordenador. Los métodos existentes son capaces de resolver el problema para determinadas figuras geométricas y en ciertas condiciones de iluminación, fondo y posicionamiento del objeto con respecto a la cámara.

- El problema de *estimación de movimiento* tiene como objetivo producir una estimación de la velocidad o posición de un objeto que se presenta en una secuencia de imágenes. Entre las aplicaciones más importantes destaca la de tracking.

En la figura 1.1 se presenta, a modo de ejemplo, el problema de estimación de la pose de un satélite en órbita. El objetivo es determinar los parámetros de pose del objeto a partir de la posición relativa de la cámara con la cual se captura la secuencia. La transformación del sistema de coordenadas del objeto al de la cámara viene determinada por los parámetros de pose 3D.

1.1. Objetivos

El objetivo principal de este proyecto es el desarrollo de algoritmos de refinamiento de pose 3D a partir de una estimación inicial. Esta primera estimación se puede obtener mediante métodos de detección que en cualquier caso no forman parte del alcance de este trabajo. Para valorar la robustez y respuesta de los algoritmos implementados, se ha realizado un estudio comparativo de las técnicas existentes en la actualidad. El método propuesto pretende aumentar la precisión de dichas técnicas, determinando el contexto aplicativo más adecuado a cada algoritmo en particular.

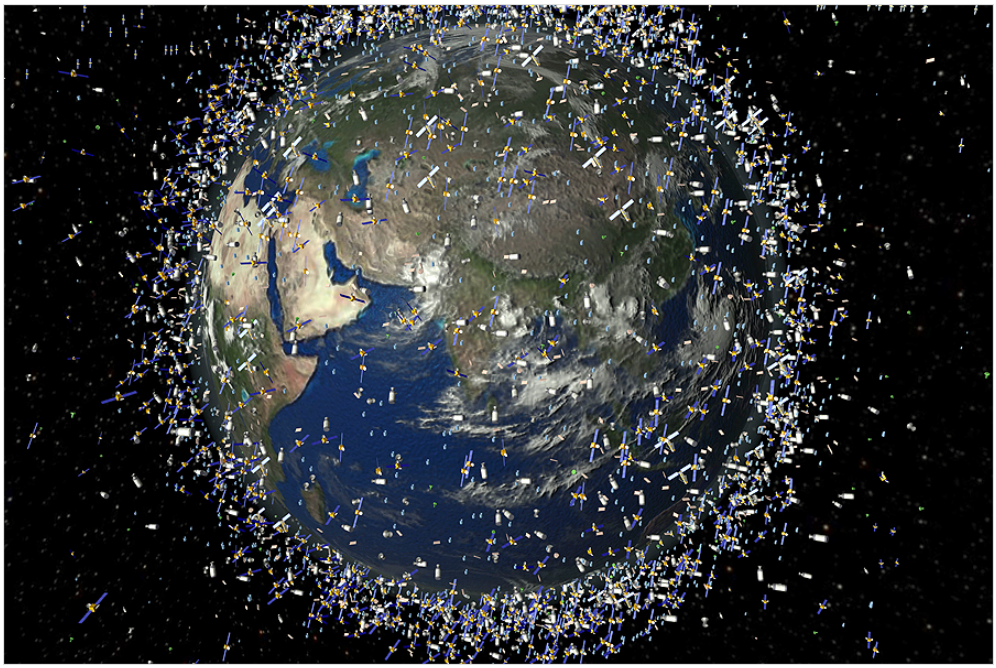


Figura 1.2: Contexto aplicativo del proyecto RISANAR. En la figura se presenta el volumen de satélites y vehículos espaciales en órbita en la actualidad.

El proceso de estimación de la pose de un objeto requiere de numerosas técnicas relacionadas con el tratamiento digital de imágenes. En este sentido es necesario realizar un análisis previo, para determinar el algoritmo que mejor se ajuste a cada fase del proyecto. Además, dado el contexto en el que nos encontramos, se precisa de una base sólida de geometría proyectiva y álgebra. En definitiva, el estudio del algoritmo de estimación de pose planteado inicialmente se constituye como un estudio general de las técnicas de tratamiento digital de imagenes, visión por ordenador y geometría proyectiva.

El estudio que se propone se engloba dentro de los proyectos RISANAR y HANDLE, desarrollados por el departamento de Ingeniería de Sistemas y Automática de la universidad Carlos III de Madrid en colaboración con otras universidades Europeas.

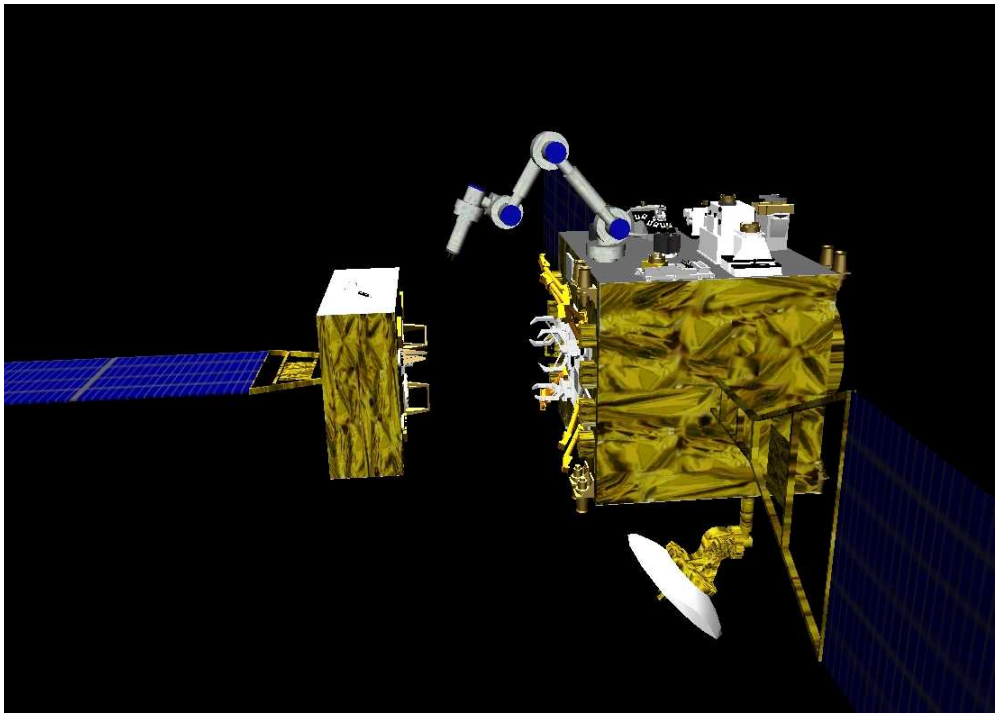


Figura 1.3: Proyecto RISANAR. Satélite de reconocimiento e inspección reparando vehículo espacial mediante brazo mecánico.

1.2. Contexto Aplicativo

El proyecto RISANAR ("Satellite Recognition and Inspection via Relative Autonomous Navigation") se aproxima al problema de la navegación autónoma de satélites en órbita. Cada año, el número de satélites y vehículos espaciales aumenta. La mayoría de ellos son capaces de cumplir con su misión, sin embargo, algunos, debido a problemas mecánicos o algún tipo de anomalía en su funcionamiento dejan de realizar lo esperado. Las consecuencias económicas son directas, así como un factor de riesgo para el resto de objetos en órbita. En este sentido, se explica la necesidad de flotar un vehículo de inspección que arregle el problema. El vehículo espacial se dota de un sistema de visión y de uno o más brazos mecánicos. El proyecto se centra en las fases de visión, detección de posición y navegación entorno al objeto estropeado. Se precisa de un algoritmo en

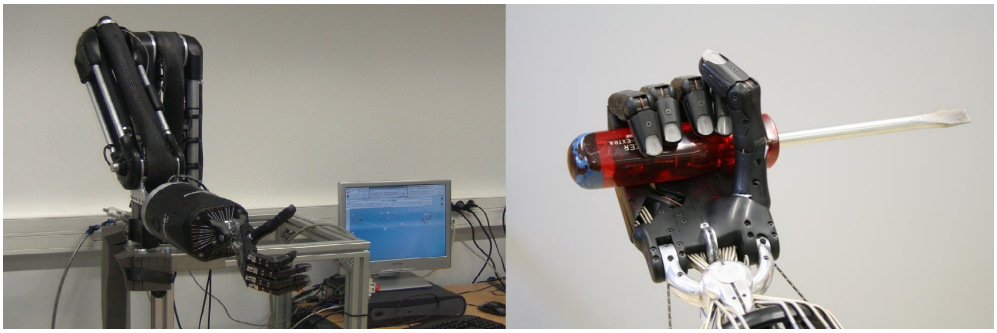


Figura 1.4: Proyecto HANDLE.

tiempo real que determine la pose 3D del satélite, así como identificar la parte o partes que están estropeadas para realizar una aproximación correcta sin dañar la estructura del objeto. El objetivo a largo plazo es incluir estas nuevas tecnologías de navegación y reconocimiento en nuevos satélites con un mayor grado de autonomía.

HANDLE se consolida como un proyecto a gran escala financiado por la UE con numerosos participantes e investigadores. El consorcio está formado por nueve socios pertenecientes a seis países europeos entre los que se encuentra la Universidad Carlos III de Madrid. El objetivo principal del proyecto es estudiar cómo los humanos realizan la manipulación de objetos, con el ánimo de replicarlo de la forma más precisa y natural sobre una mano artificial articulada. En este contexto, es de vital importancia dotar al robot de un sistema de visión que le permita reconocer y determinar la pose del objeto que desea manipular. Además, es fuente de información necesaria para conocer su estructura y así poder manipularlo.

El informe se estructura en tres grandes bloques. El primero de ellos recoge las bases teóricas, realizando un repaso de las técnicas relacionadas con la estimación de pose (capítulos 2, 3, 4 y 5). En el segundo bloque se plantean los algoritmos implementados evaluándose sus prestaciones y principales características (capítulos 6 y 7). Por último, se presentan las conclusiones más significativas del trabajo realizado y se proponen las líneas de investigación futuras del proyecto (capítulo 8).

Parte II

Estado del arte

Capítulo 2

Geometría Proyectiva

2.1. Introducción

La representación algebraica y geométrica de la realidad ayuda en muchas a ocasiones a resolver de forma más sencilla y eficaz problemas geométricos. En el campo de la visión por ordenador, la geometría proyectiva permite representar matricialmente transformaciones geométricas en el mundo 3D, caracterizando dichas proyecciones sobre el plano imagen mediante una cámara.

Con el objetivo de realizar una aproximación directa al contexto de desarrollo del proyecto, se procede a exponer los conceptos de geometría y transformaciones proyectivas 3D. Para un análisis más intenso y completo sobre las bases que rigen dichos modelos así como los conceptos relativos a la geometría 2D, ver [2].

2.2. Transformaciones proyectivas 3D

En esta sección se presentan las transformaciones geométricas de un sistema cartesiano $3D$. Un punto en el espacio se define por las coordenadas (X, Y, Z) y un pixel en la imagen por el par de coordenadas (x, y).

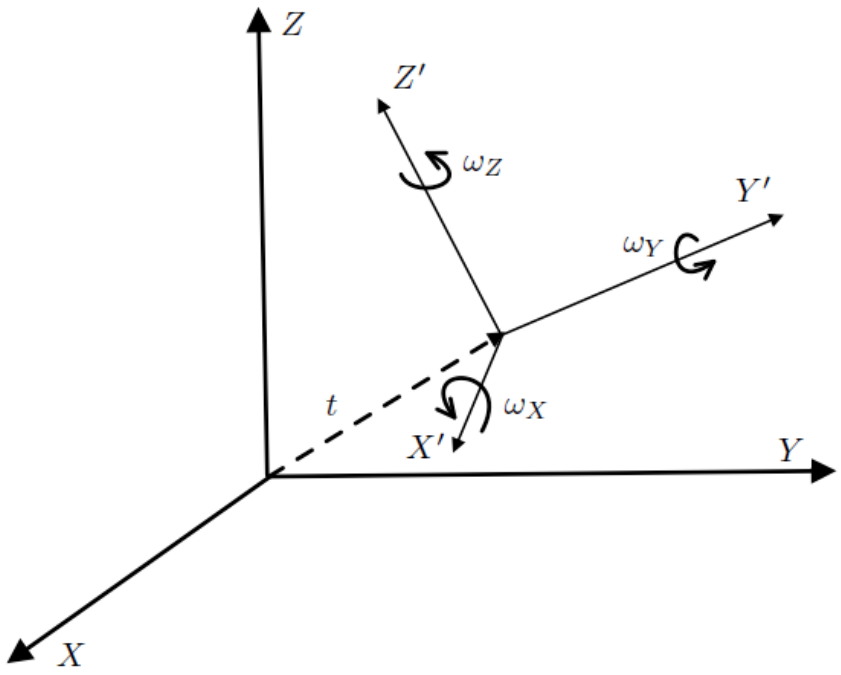


Figura 2.1: Transformación 3D euclídea.

2.2.1. Transformaciones geométricas

En esta sección se introducen las transformaciones geométricas $3D$ fundamentales para el desarrollo del proyecto: rotación y translación. Transformaciones tales como la escala no serán utilizadas ya que trataremos con *objetos rígidos*, y por lo tanto no se incluyen en esta sección. Para más información referente a transformaciones geométricas 3D, consultar [2].

En la figura 2.1 se presenta un sistema de coordenadas $3D$ (X, Y, Z) que ha sufrido una transformación de translación y rotación. El sistema nuevo generado como resultado de dicha transformación es el definido por (X', Y', Z'). La ecuación que relaciona ambos sistemas de coordenadas es la siguiente:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

donde t es un vector 3×1 que identifica la translación y R es una matriz 3×3 que representa la rotación del sistema de coordenadas.

2.2.1.1. Rotación

En el escenario $3D$, la matriz R se define de manera particular para cada uno de los ejes. De esta manera, una transformación de rotación se descompone en tres subrotaciones, cada una correspondiente a cada eje como sigue:

$$R_z = \begin{bmatrix} \cos(\omega_Z) & \sin(\omega_Z) & 0 \\ -\sin(\omega_Z) & \cos(\omega_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_Y = \begin{bmatrix} \cos(\omega_Y) & \sin(\omega_Y) & 0 \\ 0 & 1 & 0 \\ \sin(\omega_Y) & 0 & \cos(\omega_Y) \end{bmatrix}$$

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega_X) & \sin(\omega_X) \\ 0 & -\sin(\omega_X) & \cos(\omega_X) \end{bmatrix}$$

Como se trata de aplicaciones lineales, las tres rotaciones pueden combinarse formando una única expresión de rotación global. Matemáticamente se expresa como la multiplicación de todas ellas:

$$R(\omega_X, \omega_Y, \omega_Z) = R_X(\omega_X) R_Y(\omega_Y) R_Z(\omega_Z) = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

donde:

$$\begin{aligned}
R_{11} &= \cos(\omega_Y)\cos(\omega_Z) \\
R_{12} &= \cos(\omega_Y)\sin(\omega_Z) \\
R_{13} &= -\sin(\omega_Y) \\
R_{21} &= \sin(\omega_X)\sin(\omega_Y)\cos(\omega_Z) - \cos(\omega_X)\sin(\omega_Z) \\
R_{22} &= \sin(\omega_X)\sin(\omega_Y)\sin(\omega_Z) + \cos(\omega_X)\cos(\omega_Z) \\
R_{23} &= \sin(\omega_X)\cos(\omega_Y) \\
R_{31} &= \cos(\omega_X)\sin(\omega_Y)\cos(\omega_Z) + \sin(\omega_X)\sin(\omega_Z) \\
R_{32} &= \cos(\omega_X)\sin(\omega_Y)\sin(\omega_Z) - \sin(\omega_X)\cos(\omega_Z) \\
R_{33} &= \cos(\omega_X)\cos(\omega_Y)
\end{aligned}$$

Es interesante notar que la matriz de rotación inversa coincide con la transpuesta: $R^{-1} = R^T$. Aplicando dicha propiedad, llegamos a la siguiente expresión:

$$R^{-1} = R^T = R_Z^T(\omega_Z) R_Y^T(\omega_Y) R_X^T(\omega_X) = R_Z^T(-\omega_Z) R_Y^T(-\omega_Y) R_X^T(-\omega_X)$$

2.2.1.2. Translación

La transformación de translación se modela a partir de un vector. De esta manera:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t$$

donde t es un vector de dimensiones 3×1.

2.2.2. Coordenadas homogéneas

Las coordenadas homogéneas permiten combinar las transformaciones de rotación y translación en una única matriz. De esta manera, se consiguen expresiones compactas que posibilitan realizar los cálculos proyectivos de manera más eficiente.

En matemáticas, y de manera más concreta en el ámbito de la geometría proyectiva, las coordenadas homogéneas representan un instrumento fundamental para describir un punto en el espacio proyectivo. Dado el punto M de coordenadas cartesianas (X, Y, Z), se definen sus coordenadas homogéneas como (kX, kY, kZ, k), donde k es una constante distinta de cero. Para el caso particular de $k = 0$, dicho vector representa una dirección.

La transformación entre ambos tipos de coordenadas es trivial y consiste en dividir los tres primeros términos de coordenadas homogéneas por el último.

En resumen, un punto en el espacio $3D$ se expresa en coordenadas cartesianas como:

$$M = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Su representación en coordenadas homogéneas es de la forma:

$$M_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

2.2.2.1. Notación matricial

Haciendo uso de las coordenadas homogéneas, la transformación presentada en la sección 2.2.1 se modela a partir de la siguiente ecuación:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

donde t y R representan los vectores de translación y matriz de rotación respectivamente. Además, podemos establecer de manera directa las ecuaciones que rigen la transformación inversa como sigue:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R^{-1} & t' \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix}$$

donde $R^{-1} = R^T$y $t' = -R^{-1}t$.

2.3. Modelo de una cámara

En esta sección se introduce la geometría relativa a la proyección de una escena $3D$ sobre imagen $2D$ mediante una cámara. Existen muchos modelos matemáticos que explican dicha transformación a partir de los diferentes fenómenos ópticos. A continuación se presenta el modelo más utilizado y popular: el modelo *pinhole*, así como su modelo asociado con parámetros de distorsión.

2.3.1. Modelo ideal pinhole

El modelo pinhole de una cámara consiste en un plano imagen R, lugar donde se proyecta la imagen y un punto C o centro óptico, lugar de convergencia del conjunto de rayos de dicha proyección. El parámetro f, o distancia focal, indica la distancia entre el plano imagen y el centro óptico. En las figuras 2.2 y 2.3 se presentan el modelo de cámara pinhole y su modelo geométrico equivalente respectivamente.

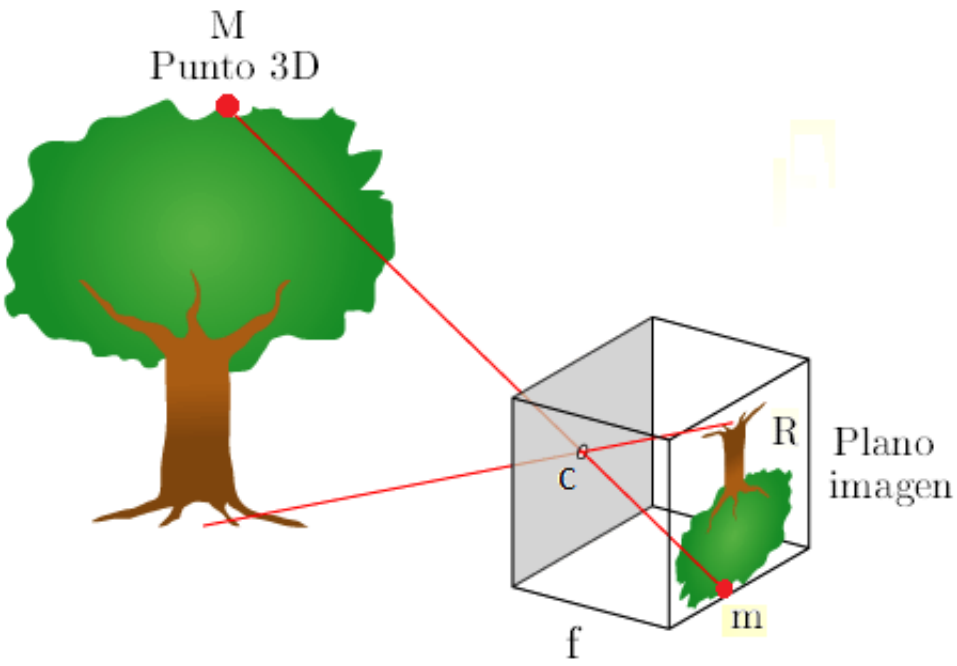


Figura 2.2: Modelo de cámara pinhole

Como se puede comprobar en la figura 2.3, el punto $3D$ M se proyecta en el plano imagen formando el punto m. Dicho punto se define como la intersección del plano R con la recta C,M (es decir, la recta que pasa por C y M).

$$m = \langle C, M \rangle \cap R$$

Aplicando el teorema de Thales podemos definir la relación entre las coordenadas no homogéneas de M y m. Suponiendo que dichas coordenadas son $M = (X, Y, Z)^T$ y $m = (x, y)^T$, obtenemos las siguientes relaciones:

$$Z_x = fX$$

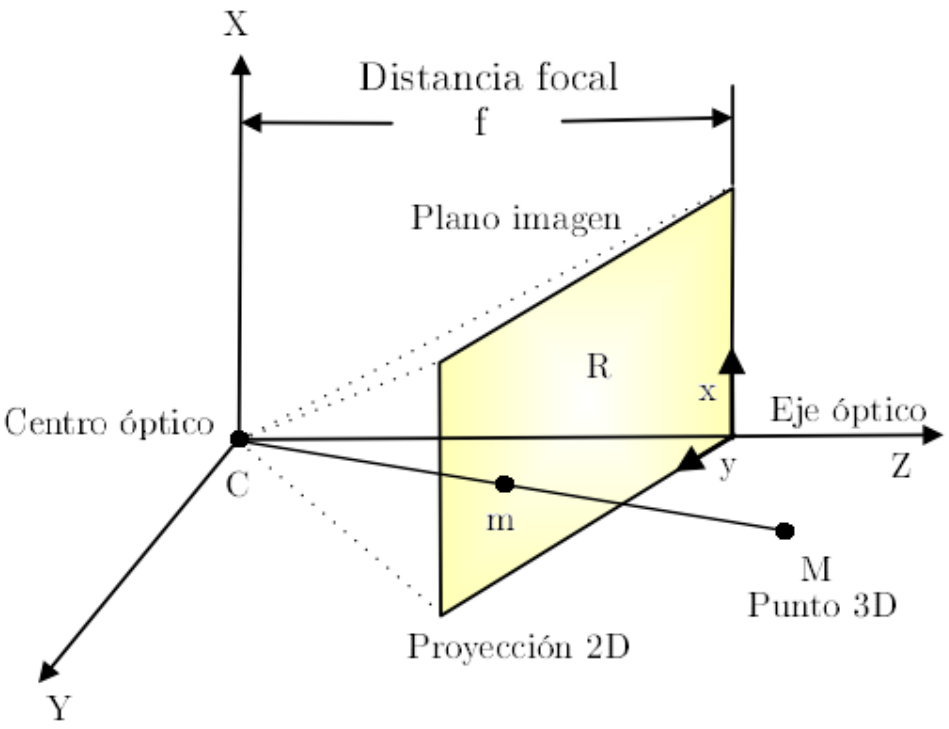


Figura 2.3: Modelo geométrico equivalente de cámara pinhole

$$Z_y = fY$$

Expresado en coordenadas homogéneas:

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

o en forma matricial:

$$\lambda m = PM$$

donde M y m representan las coordenadas homogéneas del punto M ($[\ X \quad Y \quad Z \quad 1\ ]^T$) y m $\left([\ x \quad y \quad 1\ ]^T\right)$ respectivamente. P es una matriz de dimensiones 3×4 conocida como matriz de proyección perspectiva de la cámara. El parámetro λ representa un factor de escala necesario para que se cumpla la igualdad, en este caso, igual a Z. Este factor representa una de las principales ventajas en el uso de coordenadas homogéneas en geometría proyectiva.

2.3.2. Modelo con distorsiones

En esta sección se presenta el proceso de transformación del espacio $3D$ al plano imagen, tomando como referencia el sistema de la cámara. La transformación viene determinada por los parámetros *intrínsecos* (figura 2.4), que representan las propiedades ópticas inherentes a la cámara. Se definen seis:

- Coeficientes de distorsión : k_1 y k_2
- Distancias focales (pixels no cuadrados en general): f_x y f_y
- Desplazamiento, offset del centro de la imagen : C_x, C_y.

En principio, la imagen se forma según el modelo pinhole sin distorsión. A continuación, se desplaza hasta las coordenadas C_x, C_y, desplazamiento inicial del centro de la imagen.

$$x' = \left(\frac{X}{Z} f_x + C_x\right)$$

$$y' = \left(\frac{Y}{Z} f_y + C_y\right)$$

En forma matricial:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}$$

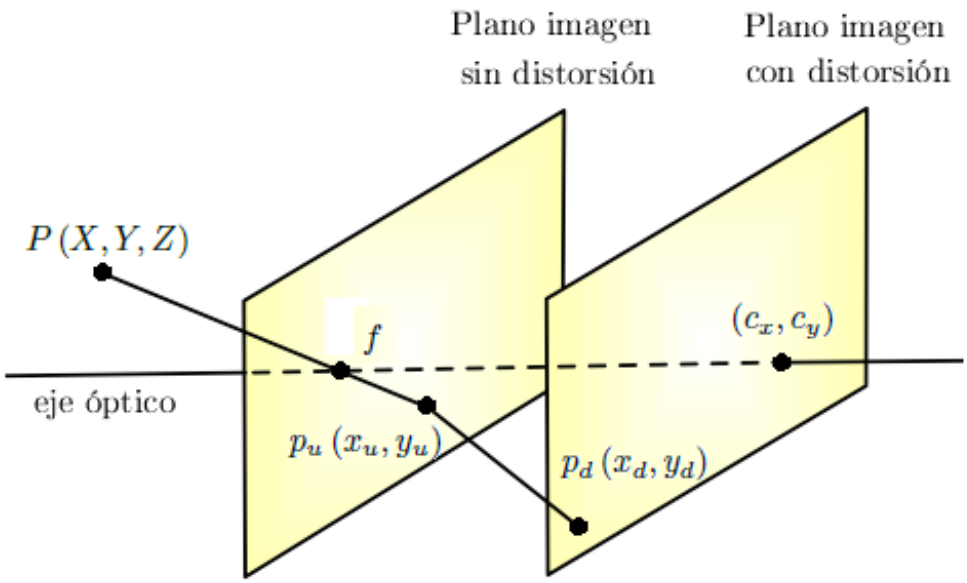


Figura 2.4: Parámetros intrínsecos en modelo de proyección con distorsión. Los puntos p_u y p_d de coordenadas (x_d, y_d) y (x_u, y_u) representan las proyecciones del punto P con distorsión y libre de ella respectivamente.

donde H se conoce como la matriz intrínseca.

En teoría, es posible definir una lente que no introduce distorsión alguna. En la práctica, sin embargo, ninguna lente es perfecta. En este sentido, es fundamental conocer los diferente tipos de distorsión que se producen en el proceso de formación de la imagen para así poder mitigarlos en la medida de lo posible. A continuación se realiza un estudio de los dos tipos de distorsión más característicos: distorsión radial, dada la forma de la lente, y distorsión tangencial producida por el ensamblado de las partes de la cámara.

Las lentes de la cámara a menudo distorsionan la localización de los pixels próximos a los extremos de la imagen. Este fenómeno es la base de los efectos "*barril*" y "*ojo de pez*" (figura 2.5). La distorsión radial es prácticamente nula en el centro de la imagen y se incrementa a medida que nos alejamos hacia las periferias. En la práctica, dicha distorsión es pequeña y puede ser modelada mediante los dos primeros términos de la serie de Fourier, k_1 y k_2. Para cámaras con alto grado de distorsión

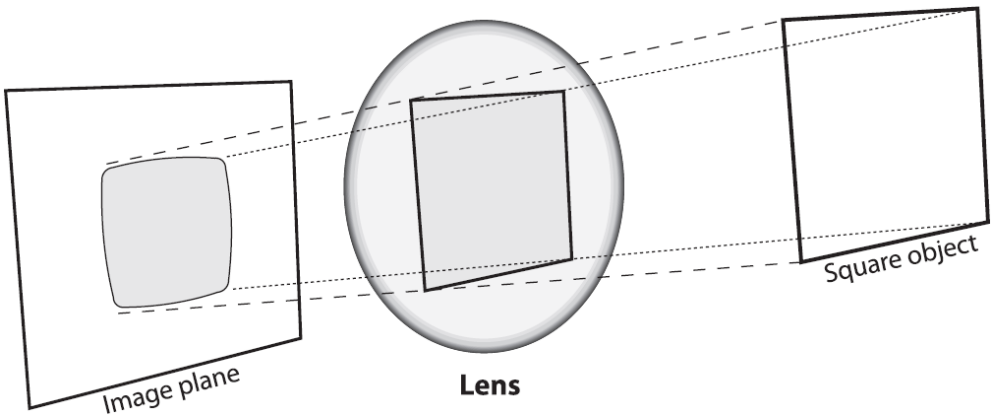


Figura 2.5: Distorsión radial. Los rayos que se alejan del centro óptico se curvan produciendo el efecto que se observa en la imagen. Este tipo de distorsión radial se conoce como *distorsión de barril*. Imagen extraída de [3].

radial, se introducirían nuevos términos k_3,k_4, etc. La ecuación que rige la localización de un punto en la imagen teniendo en cuenta este tipo de distorsión, es de la forma:

$$x_{corregida} = x' \left(1 + k_1 r^2 + k_2 r^4\right)$$

$$y_{corregida} = y' \left(1 + k_1 r^2 + k_2 r^4\right)$$

donde x e y son las coordenadas originales en la imagen y $x_{corregida}$ e $y_{corregida}$ representan las nuevas coordenadas libres de distorsión radial. La figura 2.6 presenta el desplazamiento que se produce en una cuadrícula rectangular debido a la distorsión radial.

El segundo tipo de distorsión más común es la tangencial, debida al proceso de construcción de la cámara y consecuencia de no tener las lentes perfectamente paralelas al plano de la imagen (figura 2.7). Este tipo de distorsión se caracteriza por los parámetros p_1 y p_2 como sigue:

$$x_{corregida} = x' + \left[2p_1 y' + p_2 \left(r^2 + 2x'^2\right)\right]$$

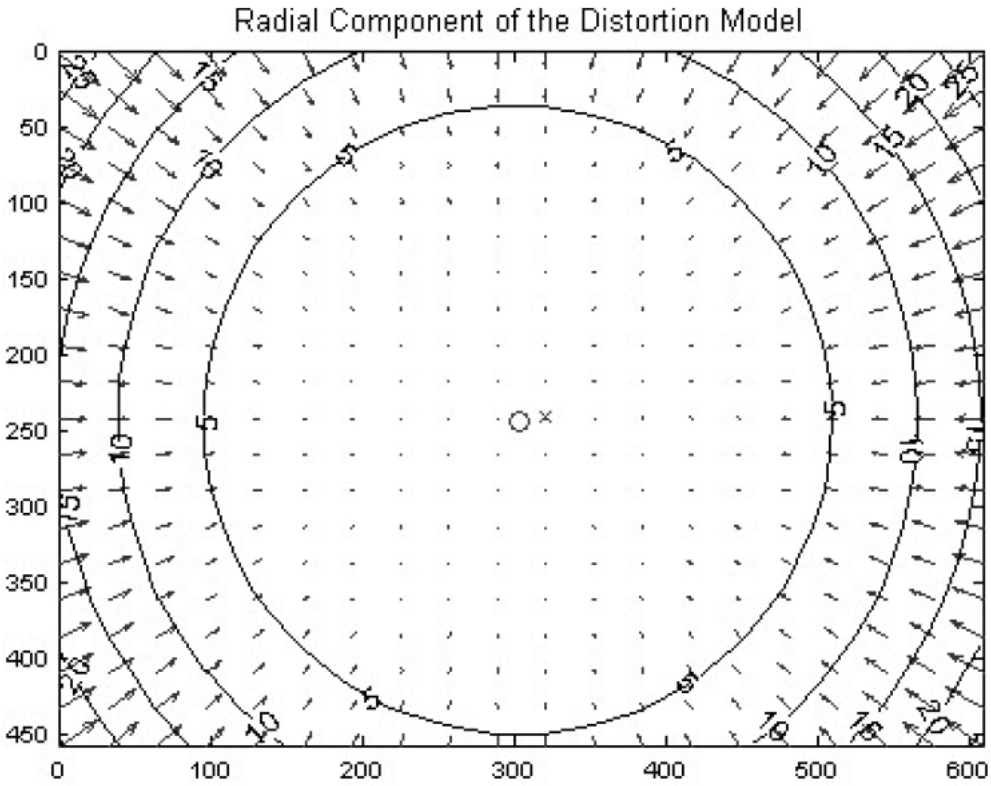


Figura 2.6: Distorsión radial sobre cuadrícula rectangular. Las flechas indican la transformación de posición de los puntos en la imagen distorsionada. Figura extraída de [3].

$$y_{corregida} = y' + \left[p_1\left(r^2 + 2y'^2\right) + 2p_2x'\right]$$

En la figura 2.8 se presenta el efecto de distorsión tangencial sobre una rejilla rectangular. Los puntos se desplazan elípticamente en función de su localización y radio.

En conclusión, la distorsión en la cámara se modela mediante seis coeficientes. Existen otro tipo de distorsiones, si bien, su impacto es reducido y no se suelen modelar.

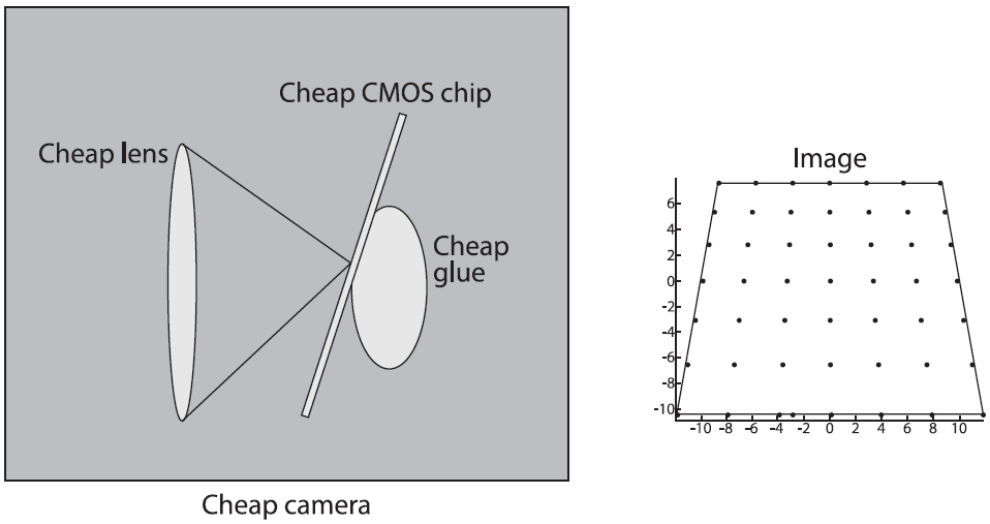


Figura 2.7: Distorsión tangencial, lentes no paralelas al plano imagen. Figura extraída de [3].

2.4. Calibración

El proceso de calibración de una cámara consiste en determinar todos aquellos parámetros que intervienen en el proceso de formación de la imagen. En general, se definen dos tipos de parámetros: geométricos (disposición de los pixels en la imagen) y radiométricos, relativos al brillo del objeto proyectado (ganancia del amplificador interno de la cámara, tiempo de exposición, apertura de la óptica). Tradicionalmente sólo se consideran los parámetros geométricos.

El proceso de calibración es fundamental para numerosas aplicaciones ya que posibilita determinar la correspondencia proyectiva entre un punto $3D$ en la escena y un punto $2D$ en el plano imagen. En el caso particular de robots móviles y manipulación (proyecto HANDLE), si el sistema de visión integrado no está correctamente calibrado, no dispondrá de información precisa sobre la posición relativa del objeto y por lo tanto será imposible manipularlo. Por otro lado, si la distorsión no se rectifica, los parámetros o características asociadas a dicho objeto, tales como la altura, área o perímetro serán corrompidas produciéndose errores de

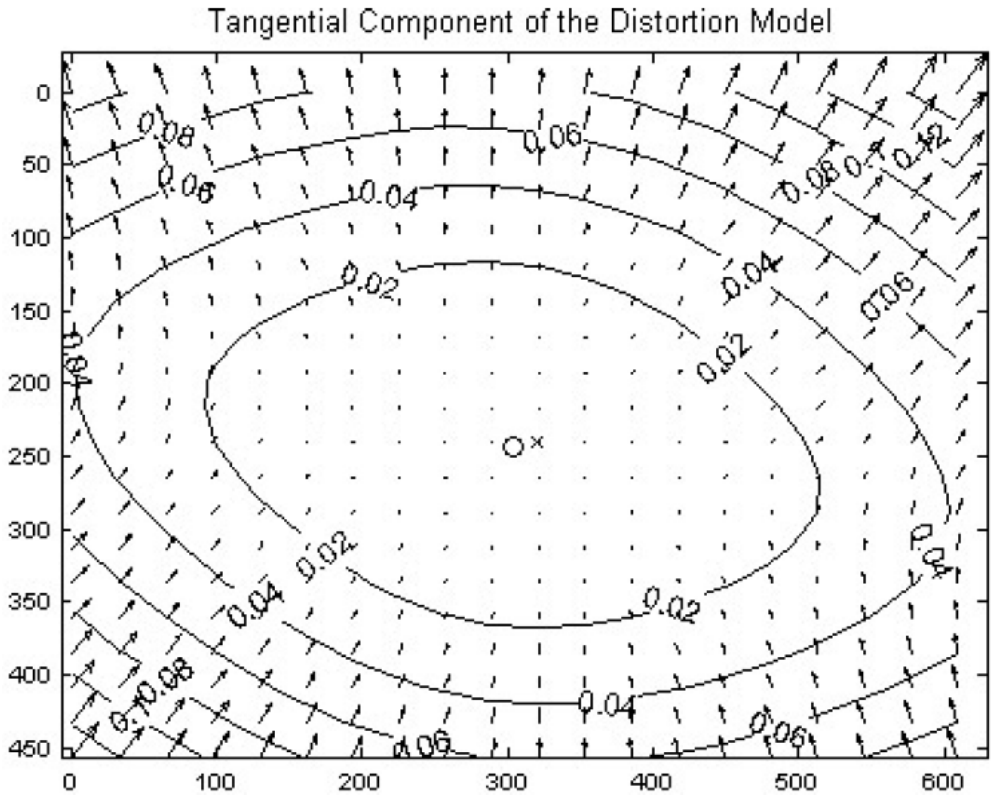


Figura 2.8: Distorsión tangencial sobre cuadrícula rectangular. Las flechas indican la modificación de posición de los puntos en la imagen distorsionada. Imagen extraída de [3].

identificación y reconocimiento.

En el caso de la visión estéreo, en el que se realiza cierta triangulación para determinar la profundidad y así poder reconstruir la imagen $3D$, la información acerca de las rectas de proyección de los pixels asociados en el par de imágenes ha de ser precisa. En este sentido, una cámara mal calibrada introduciría múltiples errores en el proceso de reconstrucción de la imagen.

En navegación pasiva se utiliza un conjunto de marcas sobre la imagen, en posiciones conocidas para determinar la posición relativa de la cámara, así como su orientación. Como cabe suponer, si existe un error en el calibrado, la navegación pasiva arrojaría resultados erróneos que

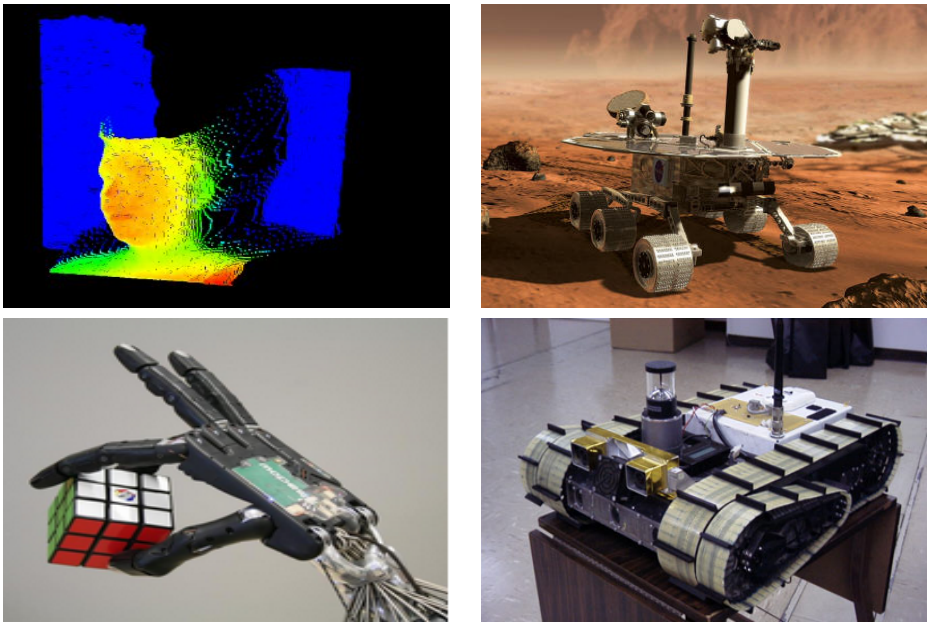


Figura 2.9: Aplicaciones que requieren calibración.

afectarían a la eficiencia y funcionamiento del sistema completo.

En ciertas aplicaciones tan sólo es necesario determinar la transformación proyectiva inversa. En otras, tan sólo la directa, como es el caso de predicción de un objeto en la imagen. En el caso particular de seguimiento o tracking donde se persigue estimar el posicionamiento continuo de un objeto móvil en la imagen, la transformación inversa permite situar el objeto (intersección plano y recta de proyección), mientras que la transformación directa avanza información sobre la posición futura del mismo.

2.4.1. Definición

El proceso de calibración de una cámara tiene como objetivo determinar los parámetros de transformación entre los puntos $3D$ de la escena y la transformación proyectiva de dichos puntos sobre el espacio $2D$ imagen. Dicha transformación se rige por dos tipos de parámetros:

- Parámetros *intrínsecos*: Estudiados en la sección 2.3.2.

- Parámetros *extrínsecos*: Es el conjunto de parámetros que determina la orientación y posición de la cámara en relación al sistema de coordenadas de referencia absoluto. Se definen los siguientes parámetros:
  - Rotación: ángulos α, β y γ.
  - Translación: T_x, T_y y T_z.

2.4.2. Procedimiento de Calibración

El proceso general de calibración presenta las siguientes etapas:

- Determinación del conjunto A, puntos $3D$ con máxima precisión.
- Identificar las proyecciones de correspondencia de A sobre la imagen $2D$, conjunto B.
- Determinar los parámetros que resuelvan la relación de correspondencia entre A y B.

Los puntos de calibración son el conjunto de puntos resultante de las dos primeras etapas del proceso. En función del algoritmo de calibración utilizado, dichos puntos cumplen o no ciertas propiedades como la coplanariedad.

En el caso de desarrollar algoritmos basados en puntos de calibración coplanarios, se suele utilizar una plantilla que implique un posicionamiento correcto. De esta manera, se realiza su identificación posterior en la imagen de forma automática. La plantilla se coloca paralela al plano X_wY_w para que el conjunto de puntos tenga igual cota Z_w. Si el algoritmo utiliza puntos de calibración no coplanarios, se establece un sistema similar en el cual se utilizan, en general, un par de plantillas situadas en distintos planos (figura 2.10).

Un sistema de calibración ha de presentar las siguientes características:

- Precisión: Un amplio conjunto de aplicaciones requieren alta precisión. En principio, el proceso de calibración debe poder satisfacer dichos requerimientos.

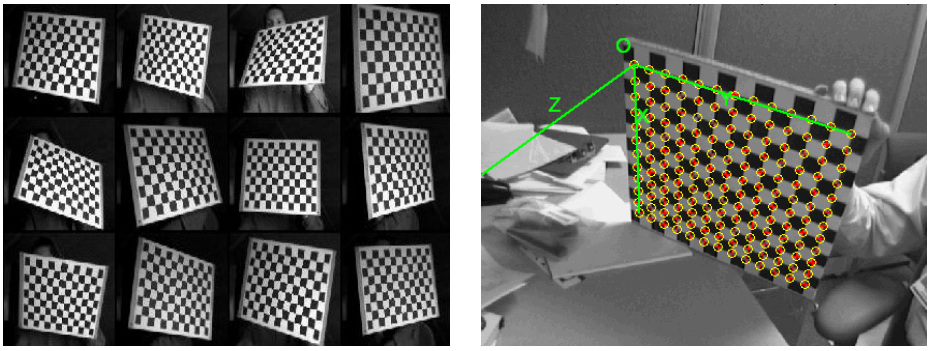


Figura 2.10: Algunas de las plantillas típicas de calibración. (a) Plantilla de puntos no coplanarios. (b) Plantilla de puntos coplanarios.

- Versatilidad: El proceso de calibración debe ser autónomo y operar de manera uniforme en un rango amplio de funcionamiento en lo que hace referencia al tipo de óptica empleado, nivel de precisión, tipo de aplicación, etc.

- Eficacia: El proceso no debería incluir subprocesos que supusiesen un coste computacional alto.

- Autonomía: El proceso de calibración debe funcionar sin la intervención de operador de cualquier tipo.

2.4.3. Principales métodos de Calibración

Una primera aproximación a la calibración nos llevaría a medir y almacenar los parámetros relativos a la recta de proyección asociados a cada pixel en la imagen, obteniendo como resultado una tabla inmensa. Sin embargo, mediante un rápido indexado seríamos capaces de determinar directamente la recta de proyección asociada a cada pixel, resolviéndose de manera precisa la proyección perspectiva inversa. La proyección perspectiva directa no sería tan fácil, ya que habría que recorrer la tabla completa hasta determinar la recta que más se ajustase al punto.

En la práctica se aplica interpolación, reduciéndose el número de pixels y dimensiones de la tabla almacenados. En el caso de obtener errores de

interpolación inferiores a los de la medida, estaríamos obteniendo precisiones similares a las obtenidas mediante el uso de la tabla completa.

Los métodos de calibración utilizan la aproximación anterior, es decir, seleccionan ciertos pixels de la imagen y calculan los parámetros relativos a la interpolación. Las diferencias esenciales entre ellos vienen determinadas por la obtención de dichos parámetros y la función de interpolación utilizada. Entre los algoritmos de calibrado más populares destacan dos: "Cálculo de la matriz de transformación Proyectiva" y el "método de Tsai".

2.4.3.1. Matriz de transformación proyectiva

Un punto w en el espacio $3D$, se observa mediante una cámara cuyo sistema de coordenadas se expresa mediante un conjunto de translaciones y rotaciones que determinan la matriz A. La proyección de dicho punto sobre el plano imagen se expresa como sigue:

$$c_h = Aw_h$$

Donde w_h representa, en coordenadas homogéneas, el punto en el espacio $3D$. Dicho punto se proyecta mediante la matriz de transformación A, de dimensiones 4×4, que a su vez es función de los parámetros intrínsecos y extrínsecos del modelo (en éste caso sólo se consideraría la distancia focal). La cuarta componente del vector resultado c_h representa un factor de escala que divide directamente a las dos primeras componentes para hacer la transformación de coordenadas homogéneas a coordenadas cartesianas. Las coordenadas (x, y) del punto sobre el plano imagen siguen la siguiente expresión:

$$x = \frac{c_{h1}}{c_{h4}} = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{41}X + a_{42}Y + a_{43}Z + a_{44}}$$

$$y = \frac{c_{h2}}{c_{h4}} = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{41}X + a_{42}Y + a_{43}Z + a_{44}}$$

Si desarrollamos la ecuación anterior, obtenemos:

$$a_{11}X_i + a_{12}Y_i + a_{13}Z_i + a_{14} - x_i\left(a_{41}X_i + a_{42}Y_i + a_{43}Z_i + a_{44}\right) = 0$$

$$a_{21}X_i + a_{22}Y_i + a_{23}Z_i + a_{24} - y_i\left(a_{41}X_i + a_{42}Y_i + a_{43}Z_i + a_{44}\right) = 0$$

Donde el punto $P_i = (X_i, Y_i; Z)$ es la representación espacial de P y $p = (x_i, y_i)$ refleja la proyección de P sobre el plano imagen, ambos conocidos. La proyección inversa resulta de la recta intersección de los planos utilizados en la ecuación anterior. Los coeficientes relativos a dicha proyección son los elementos de la matriz A.

El proceso de calibración consistirá en determinar los 12 elementos que forman dicha matriz. En este sentido, se plantea un sistema matricial de 12 ecuaciones que utiliza un mínimo de seis puntos. El sistema de ecuaciones que se define es homogéneo, de forma que existen infinitas soluciones. Por esta razón, se fija uno de los parámetros, $a_{44} = 1$ por ejemplo. En general, se utiliza un mayor conjunto de puntos de forma que se establece un sistema sobredeterminado que termina resolviéndose por mínimos cuadrados.

Si disponemos la ecuaciones anteriores en forma matricial, ajustando $a_{44} = 1$:

$$QR = S$$

donde

$$Q = \begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -(x_iX_i) & -(x_iY_i) & -(x_iZ_i) \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -(y_iX_i) & -(y_iY_i) & -(y_iZ_i) \end{bmatrix}$$

$$R = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{21} & a_{22} & a_{23} & a_{24} & a_{41} & a_{42} & a_{43} \end{bmatrix}^T$$

$$S = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Tomando como referencia n puntos, obtenemos el siguiente sistema de ecuaciones:

$$WX = C$$

Donde W es una matriz de dimensiones $2n \times 11$ formada a partir de los puntos de calibración seleccionados. X representa el vector 11×1 de incógnitas y C es el vector de dimensiones $2n \times 1$ que contiene el conjunto de coordenadas asociados a los puntos proyectados.

Se define el vector error como sigue:

$$E = WX - C$$

Aplicando mínimos cuadrados, llegamos a la siguiente solución:

$$min_X \left(E^T E\right) = min_X \left((WX - C)^T (WX - C)\right)$$

Derivando e igualando a cero obtenemos el mínimo:

$$\frac{\partial}{\partial X} \left((WX - C)^T (WX - C)\right) = 2 \left(W^T WX - W^T C\right) = 0$$

Por lo tanto:

$$X = \left(W^T W\right)^{-1} W^T C$$

Finalmente, sabiendo X, podemos reconstruir la matriz A y determinar los parámetros del modelo.

Éste método utiliza técnicas de optimización lineal, siendo ampliamente utilizado por su simplicidad.

2.4.3.2. Método de Tsai

Algoritmo propuesto por Roger Y. Tsai en 1987 [17] es en la actualidad uno de los algoritmos de calibrado más popular e implementado. Sus principales características son las versatilidad y precisión. Está constituido por cuatro etapas cuyo resultado final es la transformación del punto $3D$ al plano imagen.

1. Transformación del sistema de coordenadas de la escena al de la cámara. Primero se produce la transformación de rotación y posteriormente la de translación, en ese orden.

2. Transformación del sistema de coordenadas de la cámara $3D$ al mundo de las coordenadas ideales, libres de distorsión en el plano imagen según el modelo pinhole. La distancia focal f es el parámetro a calibrar.

3. Transformación del sistema de coordenadas de la imagen ideal al sistema de coordenadas distorsionado en el sensor. En este caso, el parámetro a calibrar es K_1 ya que K_2 se considera despreciable.

4. Transformación del sistema de coordenadas de la imagen con distorsión (x_d, y_d) al sistema de coordenadas en el computador (x_f, y_f).

El método de Tsai es el punto de referencia hacia nuevos algoritmos más robustos y flexibles. Entre ellos destaca el algoritmo propuesto por Zhengyou Zhang [29], implementado en MATLAB por Jean-Yves Bouguet [30]. Este último método es el seleccionado para realizar el proceso de calibración en el proyecto.

Capítulo 3

Visión 3D

3.1. Introducción

En este capítulo se introducen las técnicas más destacadas en visión artificial para obtener la información de profundidad asociada a una escena. Como primera aproximación se realizará un estudio teórico sobre la visión estéreo y se presenta el escáner $3D$. Por último, se analiza la tecnología ToF, técnica reciente que esta ganando popularidad y que representa una forma robusta de obtener la información de profundidad, entre otros parámetros.

3.2. Visión estéreo

En esta sección se introduce el concepto estéreo, que refleja la disposición de varias vistas sobre la misma escena para generar la visión. Existen relaciones matemáticas de tipo algebraico y geométrico que permiten relacionar dichas vistas y de esa forma crear de manera formal el conjunto imagen $3D$.

A continuación se realiza el estudio de visión estéreo bifocal.

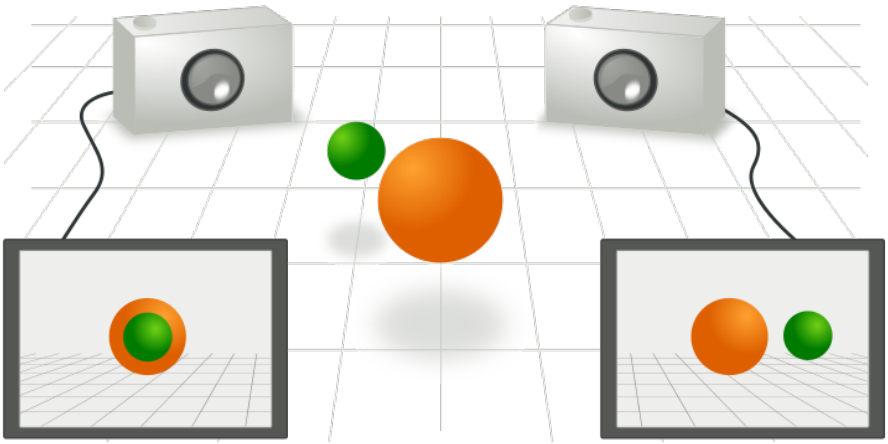


Figura 3.1: Visión estéreo. La geometría epipolar describe la relación entre las dos imágenes.

3.2.1. Análisis bifocal

La visión bifocal consiste en la utilización de dos cámaras, es decir, un par de vistas sobre el mismo objeto para construir la imagen completa. Otras modalidades incluirían una única cámara que toma imágenes sobre el objeto con cierta diferencia de tiempo y perspectiva entre cada toma.

En general, la geometría bifocal (figura 3.1) se conoce como geometría epipolar. Etimológicamente, epipolar proviene del griego epi, que referencia a lo que está encima, los polos, puntos pertenecientes a la esfera que son atravesados por el eje de rotación de la misma. A cada una de las imagenes se le asocia un epipolo como se verá mas adelante.

En la figura 3.2.a se puede observar como el punto $3D$ M se proyecta sobre el par de imagenes formando los puntos m_1 y m_2. Teniendo en cuenta únicamente uno de los puntos, por ejemplo m_1, es imposible reconstruir la posición $3D$ del punto M ya que falta la información de profundidad. Lo único que podemos afirmar es que el punto M se encontrará en la línea recta que une el punto m_1 y C_1, es decir, el punto proyección y el centro óptico asociado (figura 3.2.b). Además, en esta figura se puede observar como otros puntos también pueden formar la proyección m_1 y por lo tanto

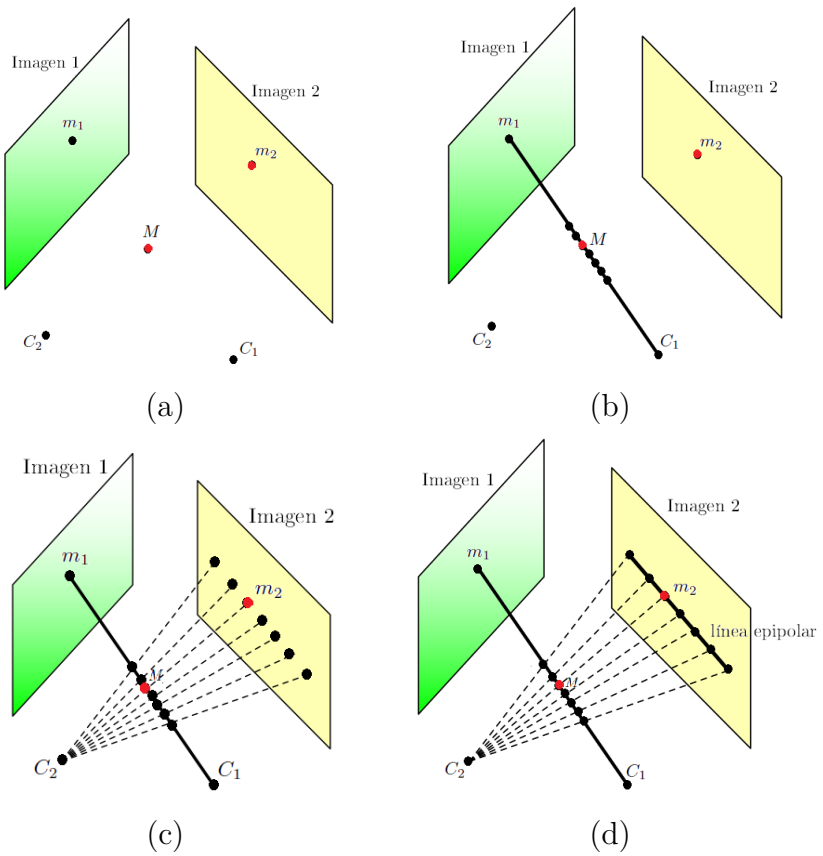


Figura 3.2: Geometría Epipolar.

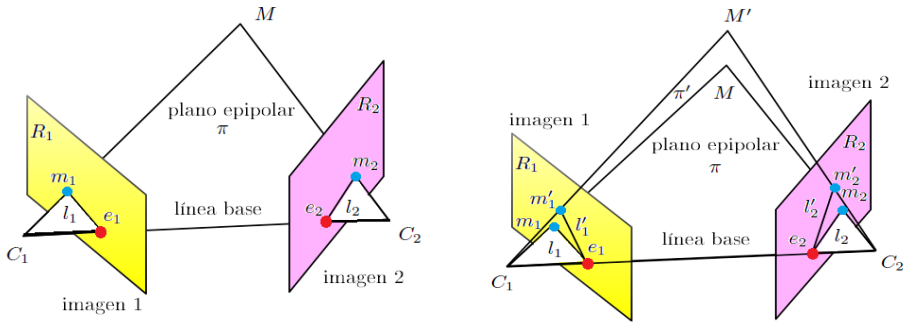


Figura 3.3: Epipolos, líneas y planos epipolares. En (a) se presentan la estructura epipolar y las líneas características. En (b) se presentan los planos epipolares.

podrían representar a M.

En la figura 3.2.c se presenta la proyección de los puntos mostrados en la figura3.2.b sobre el plano imagen 2. Se forma una recta proyectada que contiene los posibles puntos M de los cuales sólo uno será el correcto. El punto m_2 pertenece a dicha recta como no podía ser de otra manera. Por lo tanto, es necesario la utilización de ambos planos para obtener una representación de M, ya que de otro modo sólo podríamos afirmar que el punto proyectado m_2 pertenece a la línea epipolar (figura 3.2.d), pero no podríamos identificar M de manera completa.

Se establece lo que se conoce como *restricción epipolar*: El par (m_1, m_2) son puntos correspondientes si m_2 está en la línea epipolar de m_1. Esta condición es fundamental y ayuda de manera directa a identificar puntos correspondientes ya que reduce el espacio de búsqueda. En términos de dimensionalidad, representa una reducción considerable de la complejidad del problema, reduciendo el espacio de búsqueda a una línea $2D$ sobre la imagen 2. Si la imagen tiene dimensión $N \times N$, la búsqueda se realizaría en media sobre únicamente N pixels.

En la figura 3.3.a se presenta una representación alternativa de la geometría epipolar. En este caso, los planos imagen se sitúan entre el punto M y los centros ópticos.

De manera similar a lo razonado en la figura 3.2, sólo tenemos la

información de que M se encontrará en el algún punto de la recta que pasa por C_1 y M. Proyectando dicha recta sobre la imagen 2 obtenemos la línea epipolar l_2. De forma paralela, obtenemos la línea epipolar l_1. Además, se constituye el plano epipolar π, formado por los puntos M y el par de centros ópticos. Dicho plano contiene los puntos m_1 y m_2 así como sus correspondientes líneas epipolares. De esta manera, se producen las siguientes relaciones.

$$l_1 = \pi \cap R_1$$

$$l_2 = \pi \cap R_2$$

En la figura 3.3.b se presenta la aparición de un nuevo punto M' que no reside en el plano π. Los centros ópticos C_1, C_2 así como R_1 y R_2 permanecen constantes. De esta manera, se forma un nuevo plano epipolar π' que contiene los puntos C_1, C_2 y M'. Aparecen dos nuevas proyecciones relativas a m_1' y m_2', proyecciones de M' sobre los planos R_1, R_2 y se establecen dos nuevas líneas epipolares l_1' y l_2', intersección del plano π' con dichos planos. Los planos epipolares π y π' comparten los centros de proyección C_1, C_2, así como la línea recta que los une (línea base). Es en este punto donde aparece el concepto de *epipolo,* punto común de todas las líneas epipolares con el plano imagen. En la figura se representa mediante los puntos e_1 y e_2:

$$e_1 = \langle C_1, C_2 \rangle \cap R_1$$

$$e_2 = \langle C_1, C_2 \rangle \cap R_2$$

Dicho de otra manera, como los epipolos (e_1, e_2) son puntos comunes a las líneas epipolares (l_1 y l_2), entonces:

$$l_1 \cap l_1' = e_1$$

$$l_2 \cap l_2' = e_2$$

3.2.2. Análisis algebraico del par de vistas

La transformación de proyección $3D$ de M sobre el plano imagen forma el punto m. Si consideramos representaciones homogéneas de dichos puntos, podemos escribir la siguiente ecuación:

$$\lambda m = AM$$

Donde A se conoce como matriz de proyección general. Dicha matriz tiene dimensiones 3×4 y es utilizada para proyectar el punto M sobre el plano imagen. En el caso particular de visión estéreo en el que tenemos un par de vistas, obtenemos dos proyecciones m_1 y m_2 correspondientes a cada imagen. Si cada imagen tiene una matriz de proyección, obtenemos el siguiente sistema de ecuaciones:

$$\lambda_1 m_1 = AM$$

$$\lambda_2 m_2 = BM$$

Donde A y B representan las matrices de proyección para cada imagen y m_1, m_2 reflejan la representación homogénea de las proyecciones m_1 y m_2. Realizando una transformación en M, obtenemos la siguiente ecuación:

$$\lambda_1 m_1 = [I\,|0]\,\tilde{M} = \tilde{A}\tilde{M}$$

$$\lambda_2 m_2 = \tilde{B}\tilde{M}$$

donde $M = H^{-1}\tilde{M}$, $\tilde{A} = [I\,|0]$ y $\tilde{B} = BH^{-1}$. Además, H es una ma-

triz de dimensiones 4×4 regular que contiene a la matriz A. Reformulando la ecuación anterior, llegamos a la siguiente igualdad:

$$\begin{bmatrix} \tilde{a_1} & x_1 & 0 \\ \tilde{a_2} & y_1 & 0 \\ \tilde{a_3} & 1 & 0 \\ \tilde{b_1} & 0 & x_2 \\ \tilde{b_2} & 0 & y_2 \\ \tilde{b_3} & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{M} \\ -\lambda_1 \\ -\lambda_2 \end{bmatrix} = Gv = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

donde $\tilde{a_i}$ y $\tilde{b_i}$ son los elementos de la fila i-ésima del par de matrices $\tilde{A}$ y $\tilde{B}$. Si m_1 y m_2 representan el mismo punto $3D$, entonces ha de existir una única solución M. Esto se produce si $|G| = 0$. Aplicando la descomposición de Laplace se llega a la siguiente expresión:

$$|G| = m_2^T F m_1 = 0$$

donde F se conoce como la *matriz fundamental* de dimensiones 3×3 cuyos elementos se denominan Tensores Bifocales. Para más información acerca de las propiedades de esta matriz consultar [4].

Aplicando la restricción epipolar (el punto m_2 ha de estar sobre la línea epipolar del punto m_1) llegamos a la siguiente ecuación para la línea epipolar l:

$$m_2^T l = l_1 x_2 + l_2 y_2 + l_3 = 0$$

con $l = [l_1, l_2, l_3]^T = F m_1$.

Por último, cabe remarcar que la matriz F es independiente de los puntos m_1 y m_2, ya que se define únicamente a partir de las matrices de proyección de A y B. En este sentido, la matriz F relaciona los centros y planos de proyección de ambas imágenes.

Desde un punto de vista práctico, el par de puntos m_1 y m_2 se determinan a partir de un proceso de correspondencia entre ambas imagenes. Existen numerosas alternativas tales como las basadas en puntos de in-

terés, correlación, etc. En este sentido, la *textura de la imagen supone una limitación* clara para la visión estéreo.

3.3. Escáner 3D

Esta tecnología realiza un barrido del objeto o escena obteniendo la información necesaria para reconstruir el modelo $3D$ asociado. En la actualidad, es la tecnología elegida para las producciones cinematográficas así como para la industria de los videojuegos. Entre otras aplicaciones, destaca por la generación de modelos $3D$ usados en visión por ordenador.

El principio de funcionamiento es sencillo. Se genera una nube de puntos a partir de diferentes perspectivas geométricas del objeto. Si se obtiene información de color, se puede realizar un mapeo a la nube de puntos, construyendo un modelo $3D$ más próximo al real. A diferencia de las cámaras convencionales, el escáner $3D$ obtiene información sobre la superficie del objeto. De esta manera, se genera un mapa de distancias a cada punto del modelo, generando un mapa $3D$ de la escena.

Existen dos grandes familias: *con contacto y sin contacto*. Entre ellas, la más popular es la de contacto, siendo la más utilizada en la actualidad.

Los escáneres de contacto se caracterizan por examinar el objeto o escena mediante un toque físico. Dentro de esta familia se encuentran las tecnologías denominadas activas y pasivas.

- Las *activas* se caracterizan por que el proceso de detección $3D$ del objeto se realiza mediante una emisión o radiación de algún tipo (radiográficas, por ultrasonidos o por luz modulada). Entre los más destacados se encuentran el escáner láser por tiempo de vuelo y el escáner láser de triangulación 3D.

    - *Escáner láser de tiempo de vuelo 3D*: La tecnología que utiliza para determinar las distancias es esencialmente un módulo ToF, explicado en la sección 3.4. La figura 3.4.a y 3.4.b presenta un par de láseres que utilizan esta tecnología. En la figura 3.4.c se muestra una toma del proceso de reconstrucción interno de la escena.

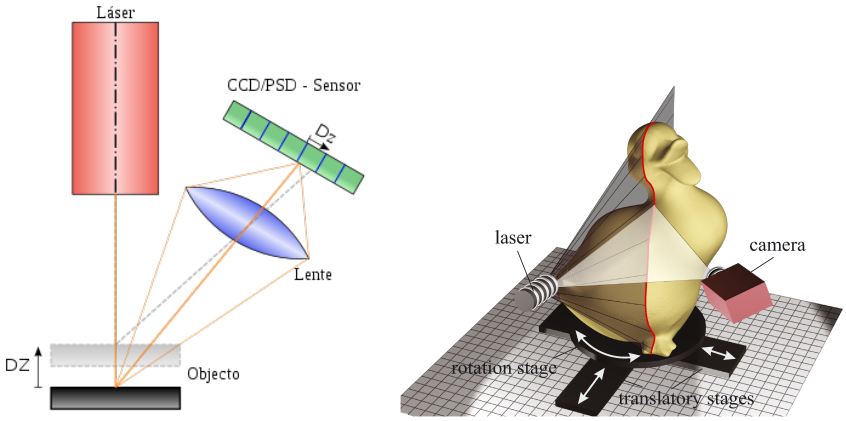
(a) (b) (c)

Figura 3.4: Escáner láser de tiempo de vuelo 3D. En (a) y (b) se presentan dos prototipos de escáneres 3D, escáneres LIDAR. En (c) se ilustra una toma del proceso de reconstrucción interno de la escena.

- *Escáner láser de triangulación 3D*: Utiliza tecnología láser para examinar la escena u objeto. El principio de funcionamiento es simple: Se hace brillar el láser en una determinada dirección. En función de la posición en la que aparezca el punto láser en la cámara se determina la distancia. Se denomina triangulación porque el emisor láser, la cámara y el punto láser forman un triángulo. En la figura 3.5.a se ilustra dicho principio. Por último, en la figura 3.5.b se presenta el proceso de reconstrucción del modelo $3D$ de un objeto utilizando esta tecnología.
- *Otros*: Escáner de holografía conoscópica, escáner de luz estructurada, escáner de luz modulada, etc.

- Los escáneres *pasivos* no emiten ningún tipo de radiación. Detectan directamente la radiación propia del ambiente en el que se encuentran. Son escáneres baratos, si bien presentan limitaciones razonables con relación a la luminosidad.

Los escáneres $3D$ presentan alta resolución y precisión. Sin embargo, es una tecnología cara y lenta (movimiento mecánico de espejos para direccionar láser en cada punto), lo que la invalida para ser utilizada en contextos de tiempo real como el que nos ocupa en este proyecto.



(a) (b)

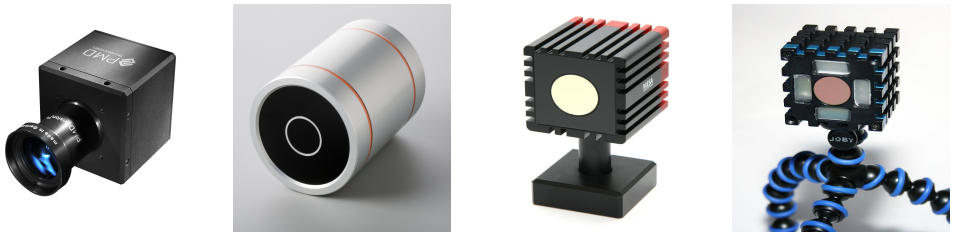
Figura 3.5: Escáner láser de triangulación 3D. En (a) se ilustra la geometría del problema y en (b) se presenta el procedimiento de reconstrucción 3D de un objeto utilizando ésta tecnología.

3.4. Cámaras ToF

La tecnología ToF (*Time of Flight*) es una técnica que ha ido ganando popularidad en el contexto tecnológico de estimación de distancias y reconstrucción de modelos $3D$. La profundidad se calcula como el tiempo que transcurre entre la emisión y recepción de un haz de luz infrarroja. La señal transmitida es un pulso modulado generalmente a $20MHz$ para que la estimación resultante sea óptima en el rango de 40 cm a 7 metros aproximádamente.

Se utiliza fundamentalmente en escáneres, aparatos de adquisición de imágenes y cámaras. De esta forma, es posible obtener imagenes o secuencias en $3D$. Aunque son dispositivos más caros que los de visión estéreo, no se requiere textura para calcular la profundidad y además, no necesitan realizar el cálculo de correspondencias entre pixels, en general un proceso muy costoso. Por otro lado, son más baratos y rápidos que los láser, al precio de menos precisión. ToF se consolida como una tecnología apropiada para la estimación, en tiempo real, de la profundidad de la escena.

Las cámaras basadas en esta tecnología son dispositivos relativamente



(a) (b) (c) (d)

Figura 3.6: Modelos de cámaras ToF. En (a) se presenta la cámara CamCube con tecnología PMD. En (b) se muestra la cámara FOTONIC-B70 desarrollada por Fotonic. En (c) se presenta el modelo SwissRanger 4000 y en (d) un modelo USB de cámara ToF desarrollado en el marco del proyecto Europeo ARTTS.

modernos. El rango de distancias que cubren oscila entre los 30 cm y los 60 metros aproximádamente, teniendo una resolución de 1 o 2 cm. Las tecnologías ToF más destacadas son:

- *Luz pulsada y contadores digitales*. El tamaño típico de imagen es 128×128 pixels. Se han conseguido rangos de hasta los $6,6$ kilómetros de distancia con haces de luz lo suficientemente pequeños. Un ejemplo de este tipo de tecnología es el detector basado en InGaAs (indium-gallium-arsenide).

- *Modulación RF y detectores de fase*. PMD ("Photonic Mixer Devices") y "Swiss Ranger" son algunos ejemplos de este tipo de tecnología. El principio de funcionamiento consiste en modular la señal de salida con una portadora *RF* para posteriormente medir la diferencia de fase de dicha señal en el receptor. Swiss Ranger proporciona rangos en torno a los 5 a 10 metros, con 176×144 pixels. PMD alcanza los 60 metros de rango.

- *Gama de sensores de imagen cerrada*. Este es el tipo de cámara ToF con más proyección en la actualidad. Ejemplos de esta tecnología son Canesta [24], 3DV Zcam [25] y Kinect [27] entre otros.

3.4.1. Aplicaciones

Las aplicaciones más destacadas de este tipo de tecnología son:

- Sistemas para automóviles. La capacidad de detectar distancias es muy útil en este campo. Existen sensores para detectar objetos cercanos a la hora de estacionar, detección para la protección de los peatones y detección de objetos cercanos para evitar colisiones. En este sentido, el automóvil es capaz de advertir una situación de peligro y frena o modifica su trayectoria.

- Robótica. El autómata conoce el escenario que le rodea y es capaz de interactuar con los elementos que aparecen en el. Existen múltiples aplicaciones tales como la navegación, reconocimiento o manipulación de objetos.

- Ocio y entretenimiento. La cámara capta el movimiento de la persona y reconoce su figura. Con ello, es posible crear un personaje ficticio que imite los movimientos que el usuario haga frente al televisor. El ejemplo más actual es el proyecto Kinect que Xbox está realizando con esta tecnología.

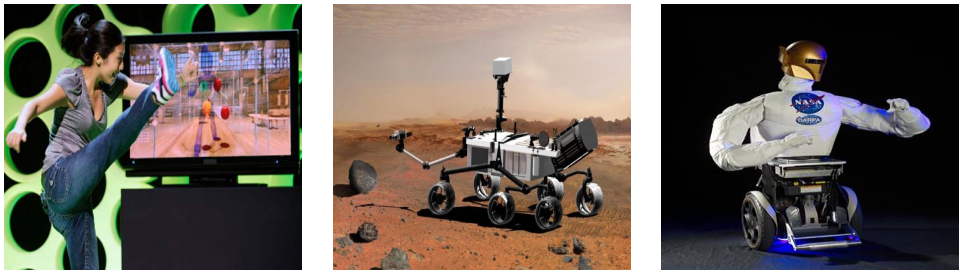
En la figura 3.7 se exponen algunas de las aplicaciones más destacadas en la actualidad para este tipo de tecnología.

3.4.2. Principio de funcionamiento

En esta sección se discute el principio de funcionamiento de la tecnología ToF así como los aspectos más relevantes asociados a cada una de las variantes que existen en el mercado.

3.4.2.1. Luz pulsada y contadores digitales

La versión más simple de ToF utiliza luz pulsada como señal modulada. Se ilumina la escena con esta señal durante un tiempo pequeño y se espera la señal reflejada por los objetos que en ella residen. En función de la distancia, el tiempo de llegada varía.



(a) (b) (c)

Figura 3.7: Aplicaciones de cámara ToF. En (a) se presenta una instantánea del proyecto Kinect de Xbox. En (b) se muestra la incorporación de cámaras ToF a un robot de reconocimiento espacial en Marte. En (c) se ilustra un robot que reconoce personas y objetos e interactúa con ellos.

El tiempo vuelo (tiempo de ida y vuelta de la señal de luz emitida) se calcula como sigue:

$$t_d = 2\frac{D_{ij}}{c}$$

donde D_{ij} es la distancia correspondiente al punto (x_i, y_j) en la imagen. La distancia asociada a cada punto se calcula como:

$$D_{ij} = \frac{t_d c}{2}$$

El rango máximo que cubre la cámara viene delimitado por t_0, ancho del pulso de la señal transmitida. La distancia máxima viene determinada por la siguiente expresión:

$$D_{max} = \frac{t_0 c}{2}$$

El circuito generador de señal se consolida como una de las partes más importantes del sistema ya que limita directamente el rango de funcionamiento. En general se utilizan láseres o LEDs especiales para reducir dicho tiempo.

Cada pixel de la cámara actúa como un fotodetector, convirtiendo la señal de entrada en corriente. En este sentido, es común el uso de

contadores digitales que paran cuando la señal deseada se detecta en el pixel correspondiente. El valor de la cuenta representa la distancia asociada a dicho pixel.

3.4.2.2. Modulación RF y detectores de fase

La técnica más utilizada en la actualidad se basa en detectores de fase. Se analiza la función de autocorrelación de la señal electroóptica utilizando cuatro muestras A_1, A_2, A_3, A_4 desfasadas 90 grados entre sí. La distancia D se determina a partir de la fase ϕ a partir de la siguiente expresión:

$$D = \frac{c\phi}{4\pi f_m}$$

donde f_m representa la frecuencia moduladora utilizada (por lo general 20MHz) y la fase ϕ es:

$$\phi = \arctan \frac{(A_1 - A_3)}{(A_2 - A_4)}$$

Además, es posible determinar los valores de amplitud, offset y escala de grises asociado a cada pixel:

$$Amplitud = \frac{\sqrt{(A_1 - A_3)^2 + (A_2 - A_4)^2}}{2}$$

$$Escala_{grises} = \frac{(A_1 + A_2 + A_3 + A_4)}{4}$$

Este tipo de sistemas eliminan las señales no deseadas que interfieren en la portadora. En ambientes exteriores, donde la interferencia es elevada, las tecnologías basadas en amplitud presentan serios problemas de funcionamiento. En este sentido, se prefiere la utilización de detectores de fase y modulación RF.

ToF es la tecnología seleccionada para desarrollar el proyecto. En el

laboratorio se dispone de una cámara ToF PMD CamCube 2.0 [26] que utilizaremos para realizar los experimentos con objetos reales.

Capítulo 4

Estimación de pose 3D

El proceso de estimación de pose $3D$ de un objeto consiste en determinar los parámetros de rotación y translación de dicho objeto a partir de información visual. El problema se consolida como uno de los retos más destacados en el campo de la visión por ordenador, surgiendo numerosas alternativas y métodos para solucionarlo. En el capítulo se presentan los algoritmos más populares y se discuten algunas de las cuestiones más fundamentales como es el número de características necesarias para el funcionamiento del algoritmo y ciertos conceptos sobre geometría proyectiva. Por último, se realizará un análisis técnico sobre el algoritmo POSIT.

4.1. Introducción

El objetivo es estimar la posición y orientación del objeto a partir de la información del modelo $3D$ y de una o varias imagenes de la escena en la que se presenta. El proceso de obtención de la imagen es variado, pudiendo ser mediante una cámara u otro tipo de sensores. La pose del objeto viene determinada por una posición (3 parámetros) y una orientación (3 ángulos) que forman un conjunto de 6 grados de libertad (6DOF).

A continuación se exponen los conceptos necesarios para la comprensión y posterior desarrollo de los algoritmos más destacados en este campo.

4.2. Datos de interés

En esta sección se introduce el conjunto de datos de interés genérico para un algoritmo de estimación de pose $3D$.

4.2.1. Datos de entrada

El planteamiento de este tipo de algoritmos se realiza suponiendo que la cámara está calibrada. Los datos de entrada al algoritmo son:

- *Modelo 3D del objeto*. En la figura 4.1 se presenta como ejemplo el modelo 3D de un satélite de comunicaciones.
- *Imagen real*. Captura real en la que aparece el objeto bajo análisis. En la figura 4.2 se presenta una imagen real del satélite de comunicaciones sobre fondo homogéneo.
- *Información de profundidad*. Este parámetro es opcional ya que depende directamente de la capacidad del sistema de visión utilizado. Los resultados que se obtienen son más robustos ya que conocemos en mayor medida la escena.

A partir de estos datos existen numerosas alternativas para resolver la pose. En la sección 4.3 se realiza una primera aproximación a dichas soluciones.

4.2.2. Datos de salida

El resultado que se persigue es la identificación de pose del modelo 3D sobre la imagen. Se plantea el siguiente sistema de ecuaciones:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix}$$

Donde:

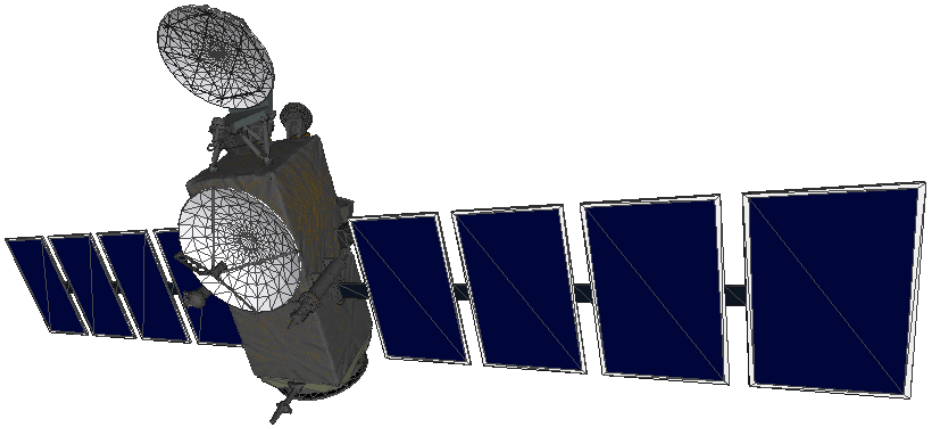


Figura 4.1: Modelo 3D de un satélite de comunicaciones.

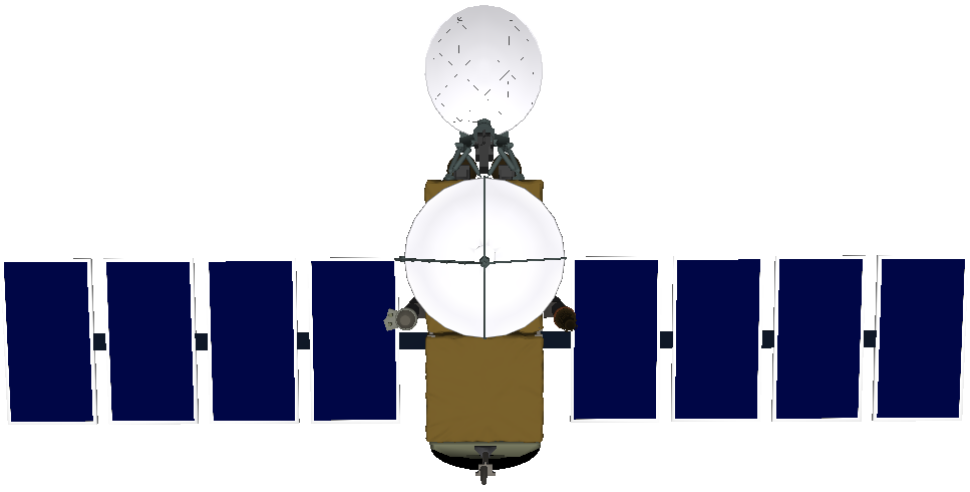


Figura 4.2: Imagen real del Satélite.

- X, Y, Z representan las coordenadas 3D de un punto del modelo.

- X', Y', Z' representan las coordenadas 3D del punto en el sistema de coordenadas de la cámara. Las coordenadas 2D sobre el plano imagen (x, y) se obtienen a partir de las ecuaciones estudiadas en la sección 2.3.2.

El objetivo principal es calcular los parámetros r_{ij} de rotación y el vector de translación $t = (t_1, t_2, t_3)^T$. A continuación se presentan las diferentes alternativas propuestas a la resolución de dicha ecuación.

4.3. Aproximaciones al problema

Existen dos alternativas a la hora de plantear la solución del problema de estimación de pose 3D a partir de los datos expuestos en la sección 4.2.1.

1. Métodos *"top-down"*. Esta alternativa tiene como objetivo el ajuste del modelo $3D$ del objeto sobre la imagen real. Es decir, se realizan diversas capturas del modelo, con diferentes parámetros de rotación, translación y escala, obteniendo un conjunto de imagenes $2D$ del objeto. El paso siguiente es identificar si alguna de dichas imágenes coincide con la real. En tal caso, la estimación de pose es directa, ya que conocemos la pose $3D$ que generó dicha imagen. Sin embargo, el coste computacional es muy alto ya que para cada pose hay que determinar si coincide con el objeto en la imagen real y así para todo el conjunto de imagenes. En este sentido, es un proceso iterativo de coste computacional elevado, lento y sensible a variaciones mínimas en la imagen.
   Además, no es adecuado para cualquier tipo de objetos, siendo óptimo para los de tipo CAD ("computer-aided design"). En las figuras 4.3 y 4.3 se presenta a modo de ejemplo el proceso de estimación de pose para un objeto de este tipo.

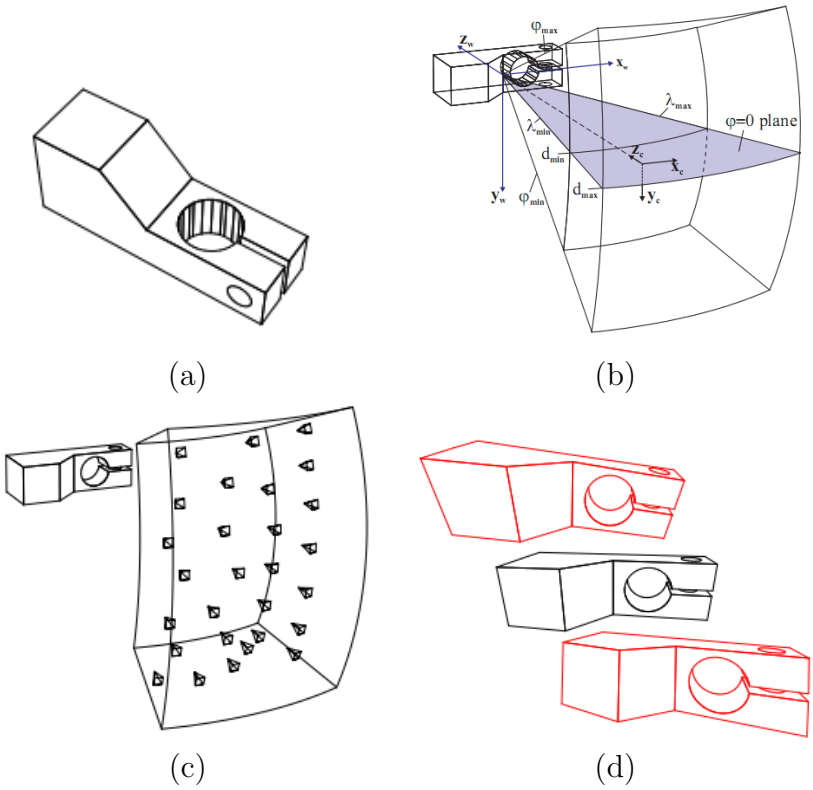


Figura 4.3: Aproximación "top-down". En la figura (a) se ilustra el modelo 3D del objeto bajo análisis. En (b) se presenta el sistema de coordenadas utilizado para determinar las diferentes vistas del objeto. En (c) observamos los puntos de vista seleccionados para generar el conjunto de imagenes de referencia. Por último, en (d) se presenta el conjunto de imágenes 2D tomadas a partir las posiciones determinadas en (c) y cuyos parámetros de pose 3D se fijan en función del sistema de coordenadas de (b). Imagen extraída de [5].

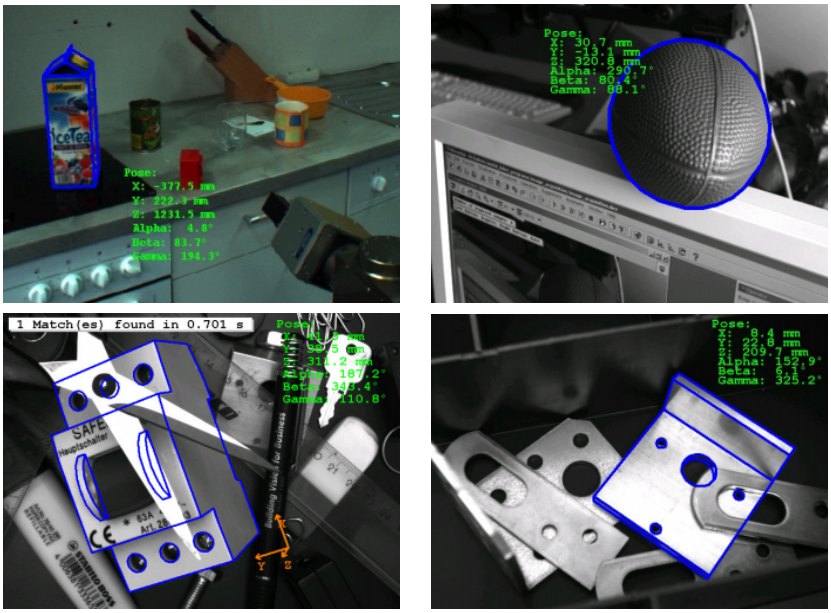


Figura 4.4: Resultados de aproximación "top-down". Imagen extraída de [5].

2. Métodos *"bottom-up"*. Esta técnica extrae características de la imagen real e intenta determinar correspondencias con la imagen modelo del objeto. A través de diferentes transformaciones matemáticas se produce un resultado de estimación 3D del objeto en la imagen. Esta aproximación es sin lugar a dudas la más desarrollada. Existen numerosas alternativas relativas a las etapas de extracción de características sobre la imagen, etapa de correspondencias y algoritmo matemático para realizar la estimación.

Bottom-up es la alternativa elegida en el proyecto para desarrollar el algoritmo de estimación de pose $3D$. A continuación se presenta un estado del arte referente a dicha alternativa.

4.4. Características utilizadas

En esta sección se introducen las diferentes características que pueden ser extraídas del modelo para realizar la optimización de pose. En la ac-

tualidad se tiende a utilizar información de puntos en la imagen que se corresponden con puntos 3D del modelo. En general, se ha tendido a esta solución por la fuerza y robustez que presentan los nuevos métodos de detección de puntos característicos (ver capítulo 5, *Puntos de interés*). Sin embargo, existen otro tipo de aproximaciones que utilizan líneas, superficies o contornos. A continuación se detalla brevemente cada una de ellas.

4.4.1. Puntos

Este tipo de característica se utiliza ampliamente en el campo de la estimación de pose 3D. Investigadores como [6, 7, 8, 9] la utilizan. El procedimiento es el siguiente: Se extraen los puntos de interés sobre el par de imagenes real-modelo y se determinan las correspondencias entre ellos. Dichas asociaciones son el conjunto de datos utilizado para el cálculo de la pose. Entre los algoritmos más importantes de extracción de puntos de interés, destacan el algoritmo de Harris, Harris-Laplace, SUSAN, SIFT y SURF.

Tomando el satélite de comunicaciones como ejemplo, la figura 4.5 presenta la extracción de puntos característicos de tipo esquina sobre la imagen.

Esta alternativa es la utilizada en el desarrollo del proyecto. En este sentido, se dedica el capítulo 5, *Puntos de interés*, a analizar en detalle las diferentes aproximaciones y alternativas a este problema.

4.4.2. Lineas

La utilización de lineas supone un método más sofisticado a la hora de realizar el cálculo de la pose 3D. Investigadores como [10, 11] las utilizan.

En realidad, la información relevante es el ángulo entre líneas y la longitud de cada una de ellas. En la figura 4.6 se presenta la extracción de líneas características sobre el modelo satélite. Como se puede comprobar, aparece información de paralelismo, perpendicularidad y longitud de cada una de ellas.

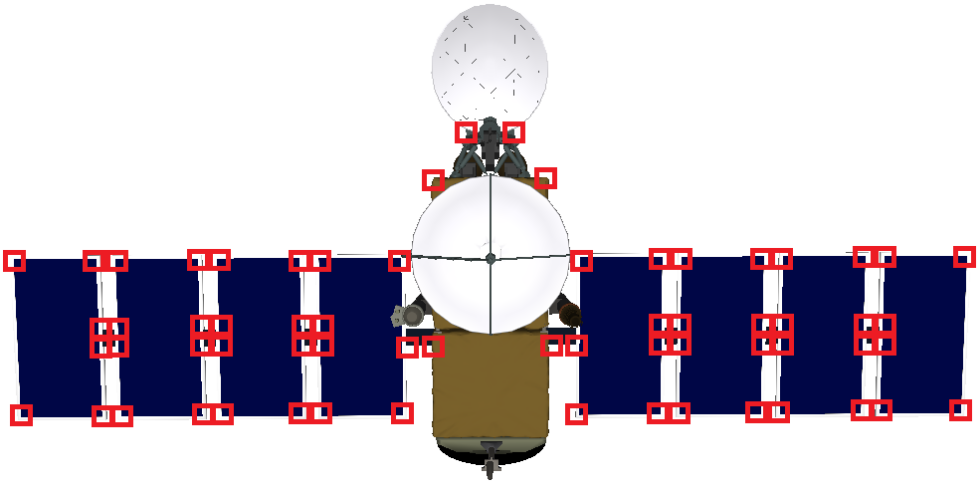


Figura 4.5: Puntos característicos de tipo esquinas sobre el satélite.

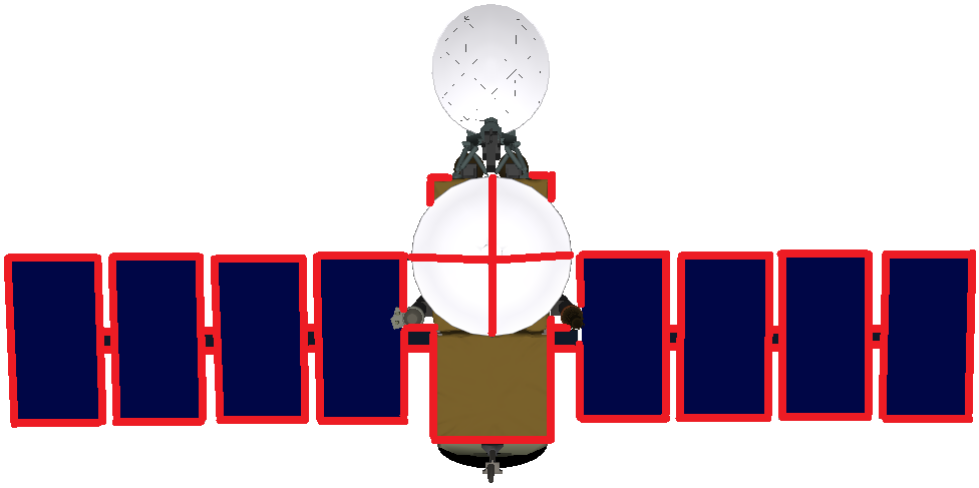


Figura 4.6: Satélite con líneas características.

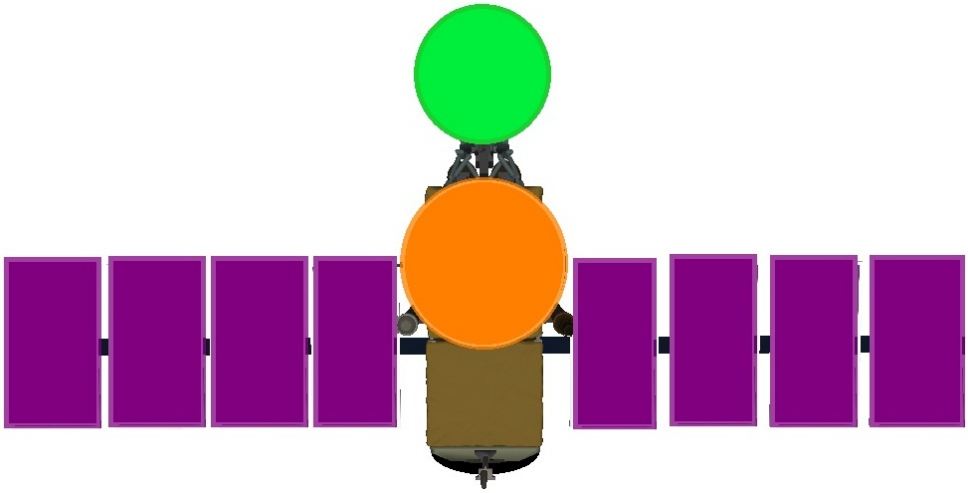


Figura 4.7: Satélite con contornos o superficies características.

4.4.3. Superficies o contornos

La utilización de contornos supone la manera más sofisticada de realizar el cálculo de la estimación de pose $3D$. En general, no se toma el contorno del objeto completo, dividiéndose en regiones de interés que son las que a posteriori se utilizan para el cálculo. Este campo de estudio es relativamente nuevo, para más información ver [12, 13].

Si tomamos como ejemplo el satélite, esto implicaría tomar de manera independiente cada una de las antenas por separado o la estructura de las alas (figura 4.7).

4.5. Estimación con puntos de interés

En esta sección se introduce el problema de estimación de pose utilizando información de puntos de característicos.

4.5.1. Numero de puntos necesarios

El número mínimo de puntos necesario para que el algoritmo converja a la solución correcta es una de las cuestiones clave para este tipo

de algoritmos. Como se ha expuesto en la introducción, se presentan 6 grados de libertad (rotación y translación, 6DOF). Cada punto introduce dos nuevas restricciones (coordenadas x e y). Por lo tanto, con 3 puntos podríamos establecer un sistema de 6 ecuaciones con 6 incógnitas y en teoría resolveríamos el reto. Sin embargo; ¿es 3 el número mínimo de puntos necesario para determinar la pose $3D$ de un objeto en el espacio?

4.5.1.1. Estimación a partir de 3 puntos

El primer investigador que se enfrenta a este reto es Lacroix en el año 1795 [15]. Sin embargo, no fue hasta el año 1981, en el que Fischer y Bolles [14] acuñaron el término "*perspective from three point problem*". Se plantearon la siguiente pregunta: ¿Bastan 3 puntos para determinar la pose 3D de un objeto? ***La respuesta es no***. Existen numerosos razonamientos matemáticos que lo prueban desde un punto de vista general.

Autores como Wolfe et al [16] han probado que existen al menos cuatro soluciones en general al problema si tomamos tan sólo 3 puntos de partida. Dichas soluciones no pueden ser ignoradas, ya que conducen a resultados erróneos. Por lo tanto, estamos obligados a tomar más puntos característicos. A continuación se presentan las diversas alternativas.

4.5.1.2. Estimación a partir de 4 puntos y 1 adicional

En general, si tomamos más de 3 puntos de partida en el algoritmo, la solución es única. La primera aproximación consiste en utilizar 4 puntos. Sin embargo, el coste computacional que implica la resolución de dichas ecuaciones ha derivado en la utilización de un punto característico adicional. De esta manera, el mínimo número de puntos es 5. En realidad con 4 puntos convergemos a una única solución, si bien el 5º punto es añadido por razones meramente de coste computacional.

4.5.1.3. Estimación a partir de 7 puntos

En este contexto destaca el algoritmo de Tsai [17], presentado en la sección 2.4.3.2 de calibración de una cámara. Este algoritmo realiza un

calibrado completo a partir de 7 pares de correspondencias ofreciendo como resultado los parámetros intrínsecos y extrínsecos de la cámara.

4.5.2. Técnicas de resolución

Las primeras soluciones algebraicas aparecen en 1841 [15], e inspiraron cierto entusiasmo entre los investigadores de la época, con resultados muy sorprendentes. En general, se definen tres tipos de metodologías en el estudio de determinación de la pose 3D de un objeto:

- Algoritmos algebraicos.
- Algoritmos de optimización.
- Algoritmos híbridos.

A continuación se presenta en detalle cada una de las variantes.

4.5.2.1. Algoritmos algebraicos

El principio de funcionamiento de este tipo de algoritmos es el siguiente: Se dispone de una fórmula que se rellena con las variables conocidas, producíendose ciertos resultados tras un tiempo de ejecución. Como el proceso se ejecuta una única vez, tenemos un resultado directo. En este sentido, si existe la posibilidad de implementar este algoritmo, la velocidad que obtenemos es muy alta.

Sin embargo, este tipo de método presenta varios puntos flacos:

- En general, el conjunto de datos disponible contiene ruido y los parámetros asociados no son exactos. Los algoritmos algebraicos requieren datos correctos, en este sentido, una pequeña variación en nuestro conjunto de datos de entrada concluiría con un resultado completamente erróneo.
- Por otro lado, este tipo de algoritmos suele ser numéricamente inestable. Es decir, se producirán resultados incorrectos si el algoritmo no se ajusta al modelo utilizado en cada caso. Los ordenadores no

disponen de espacio infinito para representar los números. En este sentido, pueden sucederse situaciones en las que dichas restricciones sean determinantes, caso de división por un número lo suficientemente pequeño que el ordenador interpretará como cero.

El término *"algoritmos lineales algebraicos"* fue acuñado por Long Quan y Zhongdan Lan en 1999 [6]. Este tipo de algoritmos utiliza el punto característico adicional al modelo $P3P$ para converger a una solución única. En la búsqueda de la determinación de la pose, término al que se refieren los autores, se define un sistema lineal al que se aplica descomposición en valores singulares para su resolución.

El conjunto de algoritmos de optimización aparece como necesidad de resolver los problemas que presentan los métodos algebraicos. Aunque en general no producen resultados exactos, su implementación en necesaria en la mayoría de los contextos.

4.5.2.2. Algoritmos de optimización

Este tipo de algoritmos también se conoce como *algoritmos iterativos*. Su característica fundamental es que no necesitan datos exactos para converger a una estimación correcta. El resultado no se calcula en un único paso. El algoritmo realiza varias iteraciones hasta que la diferencia del resultado actual y el de la etapa anterior cae por debajo de un determinado umbral, fijado al comienzo del método.

Este tipo de algoritmo tiene varios factores críticos:

- Depende de manera directa de la inicialización de la pose que se tome, estimación inicial. En este sentido, si la estimación inicial es aceptable, el algoritmo converge. Si no, el algoritmo puede llegar a ofrecer resultados completamente erróneos.

- Si el algoritmo no converge iterativamente, es decir, la diferencia entre los resultados obtenidos en las etapas i e $i-1$ crece, entonces estamos ante un problema de divergencia ante el que no es capaz de sobreponerse.

- El tiempo computacional es más alto que el de los algoritmos algebraicos debido a la reiteración y repetición del método. Si bien, en la mayoría de los casos, es necesario una aproximación de este tipo.

Un ejemplo de este tipo de algoritmos es Levenberg-Marquardt, utilizado como etapa de optimización en el desarrollo del proyecto.

4.5.2.3. Algoritmos Híbridos

Los algoritmos híbridos son una mezcla de los algoritmos de optimización y los algebraicos. De esta manera, toma las ventajas de ambos métodos. Son algoritmos numéricamente estables y robustos a nivel computacional (algoritmos de optimización) y rápidos (algoritmos algebraicos).

Un ejemplo de este tipo de algoritmos es POSIT, desarrollado en la siguiente sección.

4.6. POSIT

Algoritmo desarrollado por Daniel F. DeMenthon y Larry S. Davis en 1992 [7]. Publicaron el artículo "Model-Based Object Pose in 25 lines of Code" que sugiere que tan sólo son necesarias 25 líneas de código para obtener un algoritmo de estimación de pose $3D$.

4.6.1. Características

POSIT es un algoritmo *híbrido*. Gran parte de su estructura se considera iterativa, si bien, el hecho de la utilización de POS ("Pose from Orthography and Scaling") para la estimación de la suposición inicial, lo convierte en un algoritmo híbrido.

POSIT destaca por su velocidad. Por ejemplo, para el caso concreto de 8 puntos característicos y 4 iteraciones, el algoritmo de Lowe es 10 veces más lento y el algoritmo de Yuan sería del orden de 25 veces.

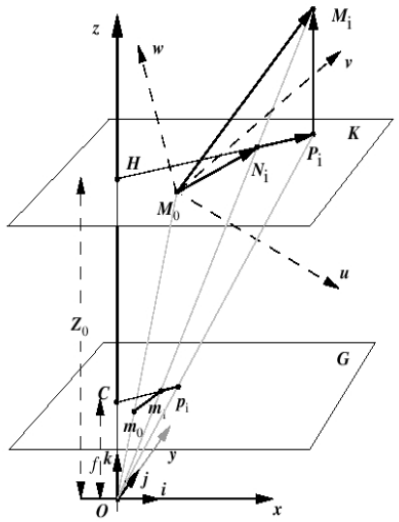


Figura 4.8: Geometría POSIT.

4.6.2. Punto de vista geométrico

En la figura 4.8 se presenta la geometría relativa al algoritmo. Los puntos M_i son los puntos característicos del modelo. M_0, que reside en el plano K, es el punto de referencia y define el sistema de coordenadas del objeto. La distancia entre el plano K y el centro de proyección O se define por Z_0. El conjunto de puntos característicos M_i representa la proyección de los puntos del modelo N_i sobre el plano K. De igual manera, m_i representa la proyección de dichos puntos sobre el plano G, plano de la imagen, a distancia f (distancia focal) del centro de proyección O, que suponemos conocido en este contexto.

Por otro lado la proyección ortonormal de los puntos característicos M_i sobre el plano K forman los puntos P_i. Además, se proyectan sobre el plano G formando los puntos p_i. Este conjunto de proyecciones se conoce por SOP ("scaled othographic projection").

4.6.3. Descripción del algoritmo

El algoritmo consta de las siguientes etapas:

1. Cálculo la matriz de coordenadas del objeto (F) a partir de los puntos del modelo M_i.
2. Cálculo de la matriz de rotación (R) a partir de los puntos imagen m_i.
3. Estimación de la pose a partir de R y F.
4. Realizar SOP y ajustar los puntos del modelo según la linea que une O y M_i ("line of view").
5. Ir al punto 2 y actualizar el valor de la matriz de rotación R.

Las fases 2 y 3 se conocen como POS. Las fases 4 y 5 dan el nombre completo al algoritmo: POS con $ITeraciones$. El método para cuando la diferencia de los resultados de la iteración t y la $t-1$ cae por debajo de cierto umbral. En general, POSIT realiza 4 o 5 iteraciones antes de converger a la solución.

En el anexo B se realiza una descripción matemática del algoritmo.

La principal *debilidad* de POSIT es la *poca robustez que presenta ante falsas correspondencias*. En este sentido, se han propuesto alternativas como SoftPOSIT ("Simultaneous Pose and Correspondence Determination"), [7], o la utilización de algoritmos como RANSAC, detallado en la siguiente sección.

4.7. Tratamiento de datos corruptos. RANSAC

En esta sección se introduce el problema de estimación de parámetros cuando las medidas están contaminados por valores atípicos, outliers. Entre los algoritmos más populares destaca RANSAC ("RANdom SAmple and Consensus"), [14]. Este algoritmo da como resultado la solución con menor error entre un conjunto de ellas que se calcula a partir de varios subconjuntos de datos de entrada elegidos aleatoriamente. *En el caso particular de utilizar puntos característicos, los datos de entrada son el conjunto de asociaciones de puntos de interés entre el par de imagenes.* El tamaño de dicho subconjunto depende de la implementación y contexto

en el que se desarrolle. Por ejemplo, en el caso de un ajuste de recta $2D$, la dimensión de dicho subconjunto debe ser al menos 3. En el contexto presente de estimación de pose, depende del algoritmo a desarrollar. Como hemos visto, existen múltiples variantes, cada una de ellas utilizando un número mínimo de puntos para su resolución.

El procedimiento es el siguiente:

1. Se realiza un ajuste del modelo con los datos hipotéticamente correctos, *"inliers"*.

2. Se utilizan los parámetros del ajuste anterior para comprobar si el resto de datos se ajusta al modelo. Si es así, se añaden dichos puntos al conjunto de datos correctos.

3. El modelo estimado se considera bueno si hay un número considerable de datos hipotéticamente correctos.

4. Se recalcula el modelo con el conjunto completo de datos correctos.

5. Finalmente, se calcula el error como la distancia entre los datos considerados correctos y sus homónimos en el modelo.

Este procedimiento se repite un número fijo de iteraciones. En cada una de ellas, se genera un modelo que puede ser rechazado por dos razones principales: número reducido de puntos en el conjunto o error relativo demasiado grande para ser aceptable. El modelo que presente menor error se guarda para posteriores comparaciones.

A continuación se exponen dos cuestiones importantes a nivel práctico: la dimensión que ha de tomar el subconjunto de datos de entrada, así como la probabilidad de obtener una solución correcta a partir de las suposiciones realizadas.

4.7.1. Selección de parámetros

Se define P como la probabilidad de obtener un subconjunto de datos correctos. Se caracteriza por la siguiente expresión:

Dimensiones de la Muestra	Porcentaje de outliers (ϵ)						
s	5 %	10 %	20 %	25 %	30 %	40 %	50 %
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Cuadro 4.1: Número de iteraciones en función de s y ϵ.

$$P = 1 - [1 - (1 - \epsilon)^s]^m$$

donde m representa el número de iteraciones, ϵ representa una estimación inicial del porcentaje de datos corruptos y s es la dimensión del subconjunto de entrada elegido.

Si tomamos logaritmos en la ecuación anterior y despejamos el parámetro m, obtenemos la siguiente relación:

$$m = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^s)}$$

En este instante se aprecia una de las características más interesantes de RANSAC: el número de iteraciones m es independiente de las dimensiones totales de la muestra de puntos de entrada al algoritmo. En la tabla 4.1 se presenta un conjunto de valores de m que producen valores de P próximos al 99 %, en función de los parámetros s y ϵ.

4.7.2. Selección del mejor resultado

El algoritmo calcula de manera iterativa el número de inliers que se obtienen con cada solución. Al concluir el número de iteraciones se escoge la solución que presente mayor dimensión.

Existen múltiples variantes y mejoras al algoritmo base. Una primera optimización del método consistiría en determinar un nuevo parámetro de ajuste p que reflejaría el porcentaje de inliers en el subconjunto. Si en la iteración correspondiente no se superase dicho valor, directamente se procedería con la siguiente iteración. Si por el contrario, el porcentaje rebasa cierto nivel, por ejemplo, el 95 %, la solución se tomaría como válida y el algoritmo finalizaría sin realizar cálculo posterior alguno.

Otras alternativas solucionan el problema de sobredimensionamiento de RANSAC tomando sólo en cuenta los inliers que se van determinando en cada iteración.

En [28] se presentan algunas de las mejoras más populares del algoritmo base.

4.8. Conclusiones

Uno de los objetivos principales de este proyecto consiste en la comparación de varios métodos de estimación de pose 3D. Como ha quedado de manifiesto, el conjunto de datos de entrada difiere de unos algoritmos a otros, si bien el resultado esperado es el mismo para todos ellos. No hay un algoritmo mejor que otro en general, siempre será referido a un caso particular o tipología referida a un determinado contexto. Algoritmos como RANSAC ayudan a obtener resultados robustos en contextos que presentan datos corruptos o erróneos.

Capítulo 5

Puntos de Interés

5.1. Introducción

En este capítulo se analizan y comparan algunos de los métodos de extracción de puntos característicos más relevantes en visión artificial.

Las características principales que ha de cumplir un *punto de interés* son:

- Ha de ser fácil de obtener.
- Unicidad. Alto nivel característico para que sea fácilmente identificable o asignable. Además, el error cometido ha de ser reducido con alto nivel de acierto en la etapa de correspondencias.
- Robustez frente a variaciones de escala, rotación, punto de vista, cambio de iluminación, color, etc.
- Fácil reconocimiento y matching entre un par de imágenes.

El proceso de extracción de puntos característicos se divide en dos etapas:

1. *Detector de puntos característicos.* Esta fase representa el punto de partida de numerosos algoritmos en el campo de la visión por ordenador. El objetivo es identificar puntos en la imagen que reunan ciertas características. La propiedad deseable para un detector de

este tipo es la repetitividad: la misma característica ha de ser detectable en dos o más imágenes diferentes de la misma escena.
La función de detección es una operación de bajo nivel de procesamiento, siendo una de las primeras operaciones que se realiza sobre la imagen. Entre los algoritmos más destacados se encuentran Harris, Harris-Laplace, SUSAN o los basados en diferencia de gaussianas.

2. *Descriptores asociados*. Una vez detectados los puntos característicos, se construyen los vectores de información asociados a dicho punto que describen su entorno local. El resultado se conoce como descriptor o vector de características. Algoritmos como SIFT o SURF combinan ambas etapas (detector de puntos de interés y descriptor asociado).

A lo largo de la historia de la visión por ordenador se han desarrollado y estudiado numerosos algoritmos de extracción de puntos característicos. A continuación se presentan algunos de los métodos más populares. Por último, se realiza una comparación entre ellos y se explican las razones por las que se elige SURF como descriptor para la realización del proyecto.

5.2. Detector de esquinas Harris

Algoritmo desarrollado por los investigadores Chris Harris y Mike Stephens en 1988 [18], se consolida como el método más utilizado en la historia de la visión por computador. En la actualidad ha dejado prácticamente de utilizarse debido a la fuerza y robustez de nuevos algoritmos como SIFT o SURF.

Los puntos seleccionados por este algoritmo son invariantes ante cambios de rotación, ruido e iluminación. Se define la matriz $C(x, y)$, calculada sobre una ventana de dimensiones $n \times n$ centrada en cada punto de interés de coordenadas (x, y).

$$C(x, y) = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \tag{5.1}$$

donde I_x e I_y representan los gradientes vertical y horizontal de la imagen respectívamente. Si consideramos los autovalores λ_1 y λ_2 de la matriz anterior, se define la matriz de autocorrelación R como sigue:

$$R = \lambda_1\lambda_2 - K\left(\lambda_1 + \lambda_2\right)$$

donde K representa un valor experimental. En la función de autocorrelación R se aprecian tres casos posibles:

- Si los autovalores de la función R son elevados estaremos ante un máximo de la función de autocorrelación. Cambios o desplazamientos en cualquier dirección producen un incremento elevado, lo que conducirá a seleccionar dicho punto como esquina.

- Si los autovalores de la función de autocorrelación R son bajos o nulos no estamos ante un punto característico, ya que variaciones en cualquier dirección produciría cambios pequeños. Por lo tanto, estaríamos ante un punto que forma parte del objeto y no representaría un borde o esquina. En otras palabras, la ventana de dimensiones $n \times n$ situada en dicho punto tiene intensidad constante.

- Si uno de los autovalores de la función R es alto y el otro es bajo estaremos ante un escenario en el que variaciones en una dirección producirán cambios pequeños y en su perpendicular cambios elevados. Por lo tanto, estamos ante un borde y no será seleccionado como punto característico.

5.3. Detector Harris-Laplace

Este método es una variante del algoritmo Harris presentado en la sección anterior. Los puntos seleccionados por Harris-Laplace son robustos ante cambios de escala y rotación. El algoritmo de detección utiliza la función de Harris, seleccionando puntos característicos sobre el espacio de escalas mediante el operador Laplaciano. En función de la escala utilizada, se dimensiona el tamaño de la región bajo estudio. En general, este

algoritmo destaca sobre el resto ya que localiza los puntos en el espacio de manera muy precisa lo que le convierte en un algoritmo deseable para tareas de reconstrucción y localización.

5.4. Detector SUSAN

Algoritmo desarrollado por S. M. Smith y J. M. Brady [19]. Al igual que Harris, es un detector de esquinas, si bien, el proceso de detección difiere de manera sustancial.

El principio de funcionamiento es sencillo: Situamos un círculo de radio fijo centrado en el pixel, definiéndose los vecinos locales de dicho punto. El pixel central se define como núcleo del conjunto y su valor de intensidad se toma como referencia. A continuación, se realiza una división de dos grupos: el grupo de pixels del entorno que tienen una intensidad similar a la del núcleo, y el grupo en el que se sitúan los pixels con intensidad diferente. De esta manera, se asocia a cada punto, un entorno local de brillo similar cuyo tamaño caracteriza la imagen en dicho punto, ya que refleja parte de su estructura. En este sentido, partes de la imagen homogéneas se verán representadas por áreas locales que ocupan la mayor parte del área del círculo.

La identificación de bordes o esquinas se realiza a partir de ciertos porcentajes. En el caso de bordes, el área cubierta del círculo sería aproximadamente el 50 %, mientras que en el caso de esquinas no rebasaría el 25 %. En la figura 5.1 podemos observar las diferencias y selección de puntos característicos para este algoritmo. Identificamos esquinas en la imagen cuando el número de pixels con intensidad similar al núcleo cae por debajo de cierto umbral.

Para dotar de mayor robustez al algoritmo, se realiza una ponderación de los pixels en función a la distancia relativa al núcleo en orden decreciente. De esta manera, un pixel cercano al núcleo tiene más relevancia que uno alejado. También se pueden definir reglas complementarias para no identificar determinados patrones u otro tipo de descriptores en el que no estemos interesados.

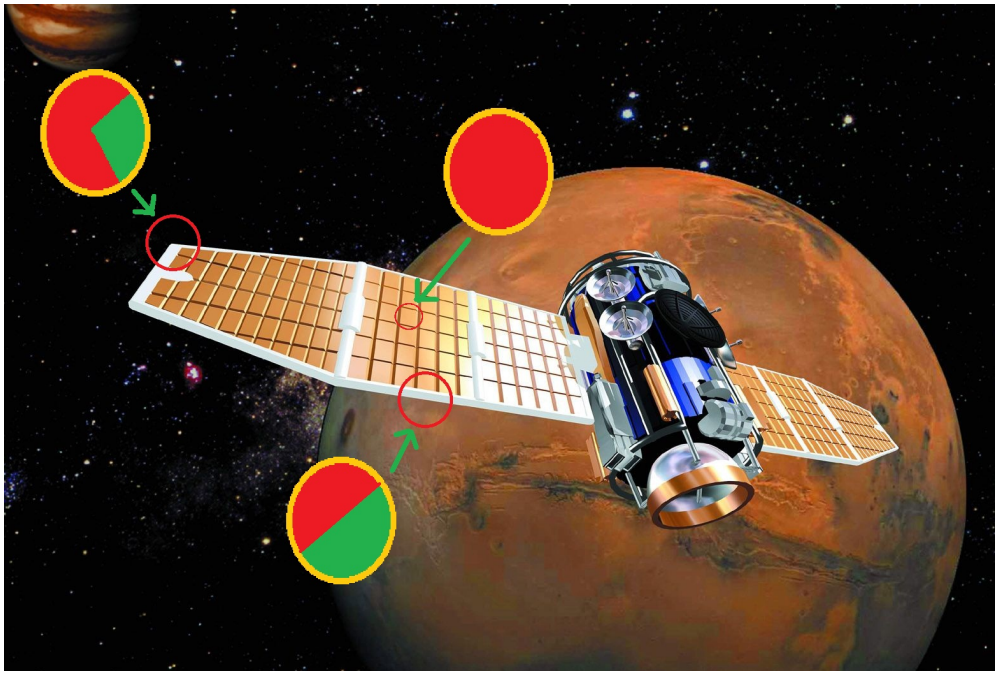


Figura 5.1: Detector SUSAN. En la figura se ilustra la diferencia entre área homogénea, esquina o borde.

5.5. Detector y descriptor SIFT

Algoritmo desarrollado por el profesor e investigador de la universidad British Columbia, David G. Lowe en el año 1999 [25]. Los puntos seleccionados por el detector SIFT son robustos ante los siguientes cambios y transformaciones:

- Ruido en la imagen.
- Cambio de escala.
- Cambio de rotación.
- Cambio pequeño de punto de vista.
- Cambio de iluminación.

Este detector se caracteriza por su buen rendimiento, precisión, tiempo de cálculo, así como por generar gran volumen de descriptores estables y robustos ante cambios.

El algoritmo se divide en cuatro etapas:

1. Identificación, en el espacio de escala, de máximos y mínimos.
2. Filtrado y localización de puntos de interés.
3. Determinación de la orientación.
4. Determinación y construcción de los descriptores asociados al punto de interés seleccionado.

A continuación se describe brevemente cada una de las etapas asociadas al algoritmo.

5.5.1. Identificación de máximos y mínimos

El proceso de detección de puntos de interés consta de varios filtrados sucesivos. El primer paso consiste en determinar la posición y escala asignables de forma repetida a diferentes puntos de vista sobre el mismo objeto. El conjunto de imagenes es filtrado mediante una gaussiana. Como resultado, los puntos SIFT son los máximos y mínimos locales que resultan de la resta o diferencia entre dichos filtros a varias escalas diferentes. Se define $L(x, y, \sigma)$ como el espacio escala de la imagen, resultado de la convolución de una gaussiana de escala variable con la imagen de entrada:

$$L(x, y, \sigma) = G(x, y, \sigma) \star I(x, y)$$

En la ecuación anterior, $I(x, y)$ representa la imagen en las coordenadas x e y. $G(x, y, \sigma)$ representa la Gaussiana de escala variable. Matemáticamente, se expresa como sigue:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\left(x^2+y^2\right)}{\sigma^2}}$$

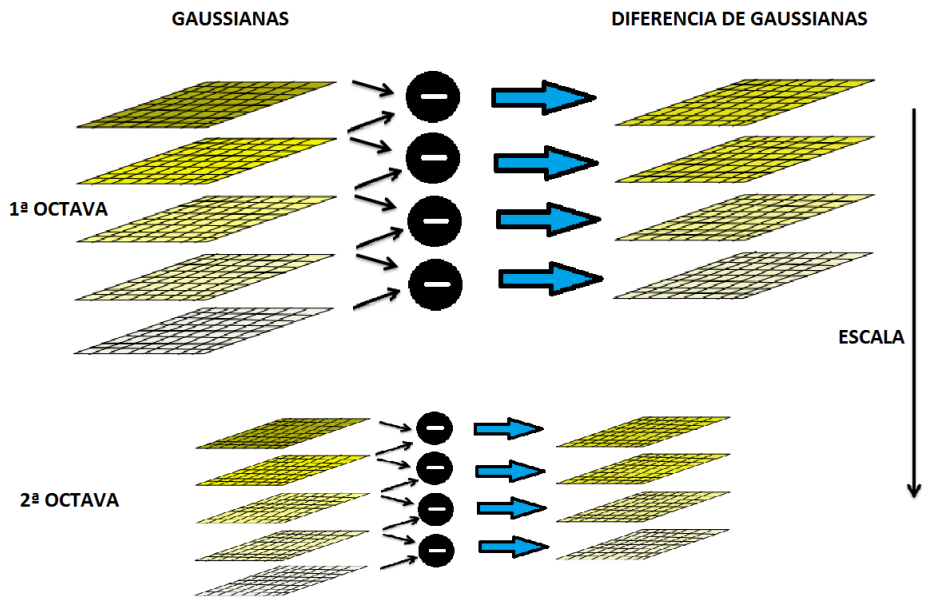


Figura 5.2: Espacio de escala DoG.

Con el objetivo de ser más eficiente y obtener puntos más estables se realiza una convolución basada en la diferencia de gaussianas en el espacio escala. Por lo tanto:

$$D\left(x,y,\sigma\right)=\left(G\left(x,y,k\sigma\right)-G\left(x,y,\sigma\right)\right)\star I\left(x,y\right)=L\left(x,y,k\sigma\right)-L\left(x,y,\sigma\right)$$

En la ecuación anterior, $D\left(x,y,\sigma\right)$ representa la convolución de la imagen de entrada con la diferencia del par de gaussianas de escala $k\sigma$ y σ. La función $D\left(x,y,\sigma\right)$ también se conoce como DoG (Diferencia de Gaussianas) y su uso implica una mejora en eficiencia, ya que únicamente consiste en la resta o diferencia de dos imagenes.

Las imagenes DoG se agrupan por octavas, divididas en cierto número de intervalos. Se define el parámetro $k=2^{\frac{1}{s}}$ como el factor de separación entre imagenes en el espacio escala. Se selecciona con el objetivo de obtener un conjunto fijo de imagenes borrosas por escala. Por otro lado, nos

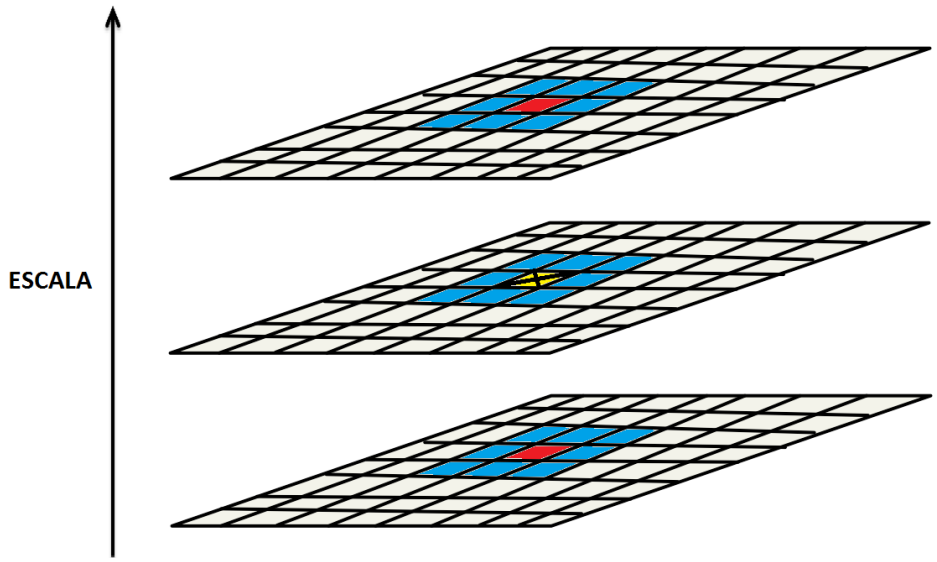


Figura 5.3: Comparación entre escalas.

aseguramos de tener un número constante de imagenes resultado de la diferencia de gaussianas igual a $s+3$. Una vez procesada cada escala por separado, la imagen gaussiana resultante se reduce en un factor dos. El proceso se repite de forma iterativa.

Los puntos clave se identifican como los máximos y mínimos locales de las imagenes *DoG* relativos a las diferentes escalas analizadas. Se compara cada pixel con sus vecinos adyacentes de la misma escala (el conjunto de vecinos suele estar fijado a ocho). Por último, se compara con los nueve vecinos de escalas próximas, tanto las superiores como las inferiores. Si el pixel central representa un mínimo o máximo, se selecciona.

Como resultado de este primer paso del algoritmo se deriva el primer conjunto de candidatos a puntos de interés.

5.5.2. Filtrado y localización de puntos de interés

En esta etapa se eliminan los puntos clave que no son adecuados, ya sea por su localización, tipología, contraste, etc. Para cada candidato, se realiza el siguiente proceso:

- Se determina su posición mediante interpolación.
- Si tiene contraste bajo se elimina del conjunto de candidatos.
- Si el punto es un borde se elimina del conjunto de candidatos.

Si el punto ha superado las fases anteriores, se procede con el siguiente paso del algoritmo en el que se le asocia una orientación.

5.5.3. Determinación de la orientación

El objetivo de esta etapa es dotar al punto de robustez e invarianza frente a rotaciones en la imagen.

El procedimiento es el siguiente: Se estima el histograma del gradiente de la orientación relativo al conjunto de puntos vecinos utilizando la imagen gaussiana de escala más próxima al punto de interés. Se define $m(x,y)$ y $\theta(x,y)$ como la magnitud y orientación del gradiente. Para cada imagen muestreada:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \arctan \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}$$

En función de la magnitud del gradiente se realiza una ponderación a cada pixel vecino y se le asocia una gaussiana de $1{,}5\sigma$ con respecto a la del punto candidato bajo análisis. Las orientaciones dominantes se corresponden con los picos del histograma. De esta manera, se crea un nuevo punto de interés con la dirección relativa al pico máximo expuesto por el histograma y con cualquier dirección que sea al menos el 80 % de dicho valor.

5.5.4. Construcción de descriptores

En esta etapa se construyen los descriptores asociados a cada punto de interés. Cada pixel candidato tiene una orientación, escala y posición que fueron asignadas en las etapas anteriores. Para crear los descriptores, el algoritmo hace uso de los histogramas de orientación. El conjunto de descriptores ha de cumplir las restricciones indicadas en la introducción de este capítulo, es decir, deben permanecer invariantes ante posibles cambios en la imagen y además, han de ser lo suficientemente distintivos como para poder ser identificados en un par de imagenes.

El descriptor se calcula a partir de los histogramas de orientación sobre una región de dimensiones 4×4 en el entorno del punto. Dichos histogramas se construyen en base a la orientación del punto principal, es decir, la imagen gaussiana de escala más cercana al punto clave. De manera paralela a lo realizado en etapas anteriores, cada punto se pondera por la magnitud del gradiente asociado y por una gaussiana $1{,}5\sigma$ del punto bajo estudio.

Como conclusión, el descriptor SIFT se constituye como un vector de dimensiones $4 \times 4 \times 8$ (8 referencias por histograma, 4 histogramas por descriptor), es decir, 128 elementos. Por último, se lleva a cabo una normalización con el objetivo de dotar al descriptor de robustez frente a cambios de luminosidad.

Resumiendo, el descriptor SIFT contiene la siguiente información:

- Posición del punto de interés (x, y).
- Escala
- Orientación.
- Información del entorno (conjunto de gradientes vecinos).

5.5.5. Matching

Una vez obtenidos los descriptores asociados al conjunto de puntos de interés en la imagen, el siguiente paso consiste en emparejar dichos

puntos con sus homónimos en una imagen consecutiva. Se hace uso de la distancia euclídea para determinar las correspondencias entre descriptores, estableciendo una asociación si la distancia relativa entre puntos es menor que k veces la distancia al vecino más próximo. El parámetro k es ajustable. En general, se fija a valores en el rango $[0{,}2, 0{,}7]$. Esta forma de realizar el matching o emparejamiento se conoce como la del *vecino más próximo*. Además, es posible aplicar otro tipo de filtrados para evitar que se realicen falsas correspondencias: filtros espaciales, angulares, etc.

5.6. Detector y descriptor SURF

Algoritmo desarrollado por Herbert Bay et al [21] en 2008. El algoritmo SURF presenta las siguientes mejoras con respecto a SIFT:

- Incremento de robustez frente a cambios o posibles transformaciones en la imagen.

- Reducción del tiempo de cálculo. Incremento de la velocidad de ejecución.

SURF introduce un nuevo descriptor más reducido que SIFT, de menor complejidad, aunque sigue manteniendo las propiedades características. El proceso por el cual se obtienen los descriptores SURF consta de las siguientes etapas:

1. Identificación de puntos de interés.

2. Determinación de la orientación.

3. Determinación y construcción del descriptor.

A continuación se describe brevemente cada una de las etapas asociadas al algoritmo.

5.6.1. Identificación de puntos de interés

SURF utiliza una aproximación de la matriz Hessiana. El motivo principal de su uso es la velocidad de cálculo así como la precisión. A diferencia de otros algoritmos, emplea el determinante de dicha matriz para determinar la posición y escala. Por lo tanto, si tenemos un punto p de coordenadas (x, y) en la imagen I, la matriz Hessiana en dicho punto a escala σ sigue la siguiente expresión:

$$H\left(p,\sigma\right) = \left[\begin{array}{cc} L_{xx}\left(p,\sigma\right) & L_{xy}\left(p,\sigma\right) \\ L_{yx}\left(p,\sigma\right) & L_{yy}\left(p,\sigma\right) \end{array}\right]$$

donde:

$$L_{xx}\left(p,\sigma\right) = \frac{\partial^2}{\partial x^2} g\left(\sigma\right)$$

De manera similar se definen $L_{yx}\left(p,\sigma\right)$, $L_{xy}\left(p,\sigma\right)$ y $L_{yy}\left(p,\sigma\right)$.

Debido a ciertas limitaciones implícitas en el uso de filtros gaussianos (aliasing, necesidad de discretización, etc), SURF implementa una alternativa basada en filtros de caja. Este tipo de filtros lleva a cabo una estimación de las derivadas parciales de segundo orden de las gaussianas involucradas. Además, se evalúan a gran velocidad permitiendo el uso de imagenes integrales, siendo transparente al tamaño de las mismas. La escala mínima se corresponde con un filtro de caja de dimensiones 9×9. Se definen D_{xx}, D_{xy} y D_{yy} como las aproximaciones de las derivadas parciales. El determinante de la matriz Hessiana se define como sigue:

$$\det\left(H_{aprox}\right) = D_{xx}D_{yy} - \left(0,9D_{xy}\right)^2$$

Gracias al uso de filtros de caja e imagenes integrales ya no es necesario aplicar de manera iterativa el mismo tipo de filtro a la salida de cada etapa. Ahora, filtros de cualquier tamaño, se aplican a la imagen original a la misma velocidad. Inicialmente se aplica un filtro de caja de dimensiones 9×9 cuya salida presenta una escala inicial $s = 1,2$ correspondiente a una gaussiana de $\sigma = 1,2$. El procedimiento consiste en aumentar de

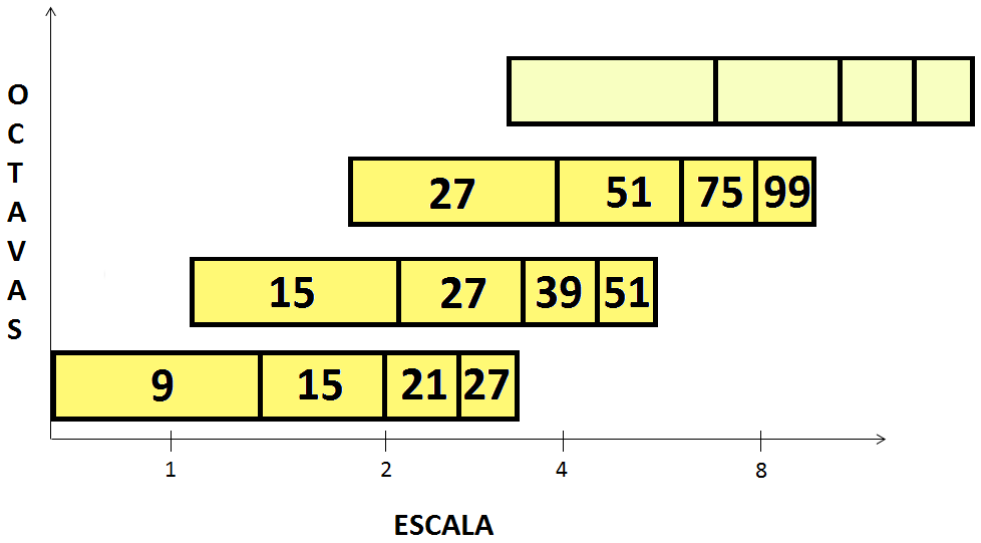


Figura 5.4: Filtros SURF. En la imagen se presenta la estructura y tamaño de los filtros utilizados en SURF a diferentes escalas.

forma iterativa el tamaño del filtro a aplicar. En cada octava se dobla el incremento aplicado en la etapa anterior.

- Octava inicial: $9 \times 9 \rightarrow^{6} 15 \times 15 \rightarrow^{6} 21 \times 21 \rightarrow^{6} 27 \times 27$
- Octava siguiente: $15 \times 15 \rightarrow^{12} 27 \times 27 \rightarrow^{12} 39 \times 39 \rightarrow^{12} 51 \times 51$
- Octava siguiente: $27 \times 27 \rightarrow^{24} 51 \times 51 \rightarrow^{24} 75 \times 75 \rightarrow^{24} 99 \times 99$
- Sucesivamente ...

Al mismo tiempo, los intervalos de muestra pueden ser doblados en el proceso de extracción de puntos clave.

De manera similar a lo realizado por SIFT, se eliminan los puntos que no sean máximos locales en una región de dimensiones $3 \times 3 \times 3$ alrededor del punto de interés. El máximo determinante de la matriz Hessiana se interpola en el espacio escala de la imagen. Como resultado tenemos el conjunto de puntos característicos.

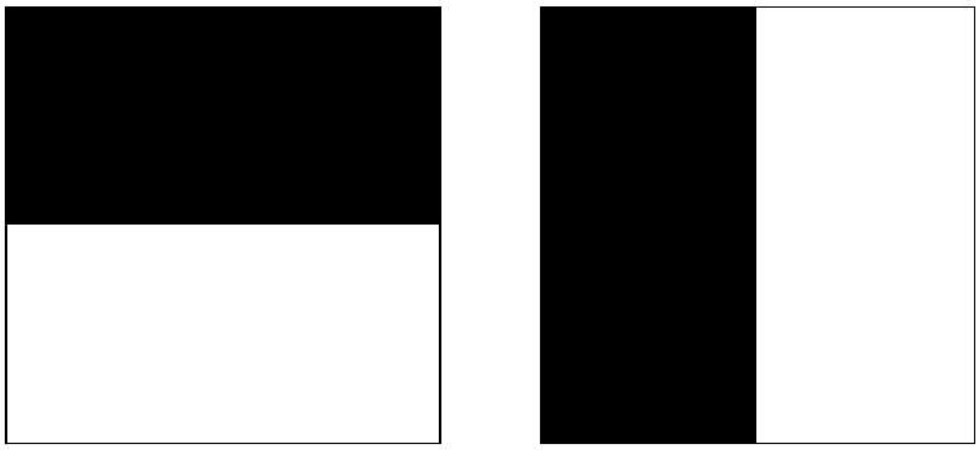


Figura 5.5: Funciones de Haar. Detector SURF.

5.6.2. Determinación de la orientación

Una vez determinado el conjunto de puntos de interés, el siguiente paso consiste en asociar a cada uno de ellos una orientación. Como resultado, se consigue que los puntos sean invariantes ante rotaciones.

El procedimiento es el siguiente: Se calcula la Respuesta de Haar con dirección x e y, ver figura 5.5, en un entorno circular de radio $6s$ (s es la escala del punto detectado) centrado en el punto de interés. El valor s también se toma como referencia para la etapa de muestreo. Las respuestas onduladas de Haar se calculan tomando como referencia dicho parámetro, es decir, a mayor escala, mayor será la dimensión de dicha respuesta. A continuación se hace de nuevo uso de imagenes integrales, consiguiéndose un filtrado más rápido. La respuesta en la dirección x e y necesita 6 operaciones.

Las respuestas onduladas resultantes se ponderan mediante una gaussiana con $\sigma = 2,5s$ centrada en el punto de interés. Dichas respuestas se representan mediante vectores en el espacio con la respuesta vertical y horizontal a lo largo del eje de ordenadas y abscisas respectivamente. La orientación dominante se obtiene mediante la suma del conjunto de respuestas en una ventana de orientación variable que cubre un determinado ángulo espacial. Es un parámetro que se calcula experimentalmente y que

permite cubrir, de manera general, $\pi/3$ radianes. Se constituye un nuevo vector como suma de las dos respuestas, vertical y horizontal. La orientación del punto de interés viene determinada por el vector de mayor longitud.

5.6.3. Construcción de descriptores

Una vez determinados los puntos de interés SURF, el procedimiento a seguir para extraer los descriptores asociados es el siguiente: se define una región cuadrada orientada según lo calculado en el paso anterior y centrada en el punto de interés. El tamaño de dicha región es $20s$. A continuación, se reduce en subregiones de dimensiones 4×4. Para cada una de ellas, se determinan las características en puntos diferenciados por regiones de tamaño 5×5. Se definen d_x y d_y como las respuestas Haar en la dirección horizontal y vertical respectivamente, referenciadas a la orientación del punto de interés bajo análisis. Dichas respuestas se ponderan mediante una gaussiana de $\sigma = 3,3s$, consiguiendo robustez frente a deformaciones geométricas o errores de posicionamiento.

Las respuestas d_x y d_y calculadas en cada subregión se suman formando la primera información del vector descriptor. Además, se suma la respuesta en valor absoluto $|d_x|$ y $|d_y|$, obteniendo información relativa a la polaridad. Por lo tanto, el vector descriptor SURF v para cada subregión tiene 4 elementos de información:

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \tag{5.2}$$

El vector descriptor resultante para las 4×4 subregiones es de longitud 64.

5.6.4. Matching

El matching de SURF es equivalente al realizado por SIFT, explicado en la sección 5.5.5.

5.7. Comparación Algoritmos

Es importante determinar el algoritmo de detección que mejor se ajusta a las necesidades del proyecto. Para ello, existen numerosos estudios comparativos entre los cuales destaca el desarrollado por los investigadores Johannes Bauer, Niko Sünderhauf y Peter Protzel de la universidad de Chemnitz en el año 2007 [22]. En dicho estudio se hace incapié en los algoritmos de detección SURF y SIFT, si bien se realiza una caracterización completa en lo relativo a otros algoritmos como Harris. El algoritmo a utilizar en este proyecto (SURF) viene determinado por los resultados de dicho estudio, ya que ofrece información acerca de la velocidad de computo, robustez e invarianza frente a cambios diversos en la imagen, número de puntos de interés extraídos, etc.

El estudio expone las siguientes conclusiones:

- El algoritmo SIFT detecta en general un mayor número de puntos característicos que SURF.
- La calidad de las asociaciones realizadas con SIFT y SURF es prácticamente igual. Si bien, es lévemente superior en SIFT.
- El detector Harris presenta un rendimiento inferior a los algoritmos SIFT y SURF en todos los sentidos.
- Ambos algoritmos (SIFT y SURF) presentan buenos resultados ante rotaciones, cambios de vista y escala con pequeños porcentajes de error.
- En lo relativo al ruido, ambos algoritmos presentan buenos resultados, con porcentajes razonables de error.
- Existe un determinado umbral de iluminación a partir del cual no se detecta prácticamente ninguna asociación. Para valores inferiores a dicho umbral el algoritmo es robusto ante cambios de luminosidad y ofrece resultados prácticamente constantes.

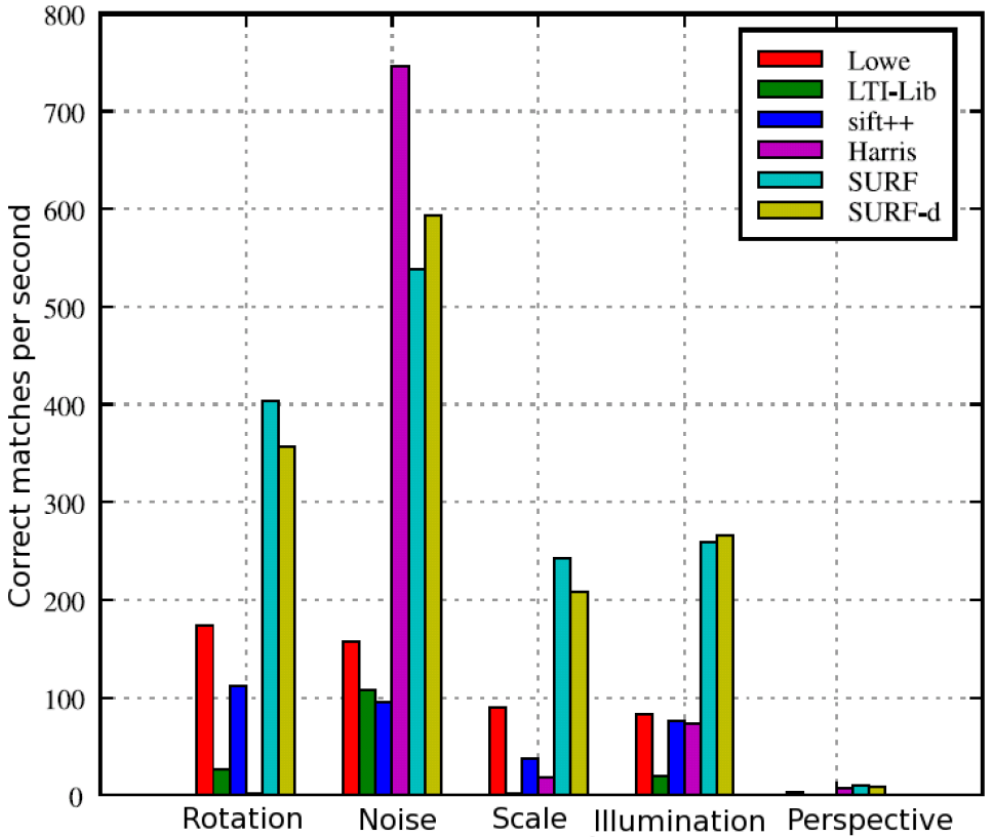


Figura 5.6: Resultado comparativo algoritmos de extracción de puntos de interés. Imagen extraída de [22].

La figura 5.6 presenta la gráfica resultado de las que se derivan las conclusiones anteriores.

Como podemos comprobar, ambos algoritmos, SURF y SIFT, presentan resultados similares. Es cierto que SIFT, a la vista de los resultados, realiza asociaciones con menor error. Sin embargo, necesita un mayor tiempo computacional. En este sentido, se elige SURF como algoritmo de extracción de puntos característicos para el desarrollo del proyecto.

Parte III

Desarrollo del proyecto

Capítulo 6

Algoritmos desarrollados

6.1. Introducción

En este capítulo se presenta la estructura general del algoritmo implementado y se analizan las principales estrategias de diseño. El planteamiento del problema comienza con la elección del tipo de algoritmo a desarrollar. A continuación, se realiza una aproximación global a los intereses y contexto aplicativo del proyecto. De esta manera, se explican las motivaciones para fijar las distintas fases y etapas que componen el método. Por último, se analiza a fondo cada una de las etapas y se proponen mejoras tales como la aplicación de RANSAC o la incorporación de un módulo de seguimiento.

6.2. Características

El algoritmo que se propone se engloba dentro de la familia de algoritmos iterativos o de optimización. Como se explica detenidamente en la sección 4.5.2.2, la característica fundamental de este tipo de métodos es que no necesitan datos exactos para converger a la solución. La aplicación de RANSAC refuerza dicha afirmación, ya que el algoritmo convergería aún habiendo un porcentaje considerable de datos corruptos.

En un caso genérico, la probabilidad de obtener conjuntos de datos con

muestras erróneas es alta, ya sea en menor o mayor porcentaje. Esta es la razón principal por la que se decide implementar este tipo de algoritmo.

6.3. Descripción del algoritmo

En esta sección se realiza la descripción general del algoritmo implementado, presentándose su estructura (figura 6.1). A continuación se analizan las etapas y características más importantes del algoritmo.

6.3.1. Datos de entrada

Existen tres fuentes de datos de entrada al algoritmo:

- *Imagen real*. Imagen en la que aparece el objeto bajo estudio. El objetivo es estimar los parámetros de pose $3D$ del objeto en dicha toma.

- *Datos fijos* (figura 6.2). Conjunto de datos previos necesarios para la ejecución del algoritmo, es decir:

    - *Modelo 3D del objeto normalizado*. Existen múltiples alternativas para obtener dicho modelo. En el caso de trabajar con objetos sintéticos, puede realizarse una modelización por ordenador con programas tipo Blender. En el caso de trabajar con objetos reales, existen numerosas técnicas de reconstrucción del modelo 3D a partir de un conjunto de imágenes del objeto. El método utilizado en el proyecto es el desarrollado por N. Burrus et al [31].
      El concepto normalizado hace referencia a que el centro de gravedad del modelo 3D debe ser el $(0, 0, 0)$, es decir, el origen de coordenadas. Si no está normalizado, se realiza una etapa inicial, previa a la ejecución del algoritmo, que normaliza el modelo. La razón fundamental es facilitar el conjunto de transformaciones geométricas que tienen lugar a lo largo de

las distintas etapas del algoritmo. Se explicará con más detenimiento en secciones posteriores.

- *Imagen de referencia 2D*. Representa la proyección 2D del modelo a partir de un conjunto de parámetros de pose 3D conocidos. En este sentido, tenemos una pose de referencia y una imagen con la cual realizaremos las diversas comparaciones sobre la imagen real. Además, se dispone adicionalmente de un *mapa de profundidad* asociado a dicha toma que se consigue proyectando el modelo 3D con la pose conocida. *Las características resultado sobre la imagen modelo son 3D*. En la figura 6.2 se presenta un ejemplo de generación de imagen 2D y mapa de profundidad a partir de un modelo y pose conocidos.

- *Pose inicial*. Estimación de pose inicial gracias a la aplicación de un algoritmo de detección o a la estimación de pose anterior (tracking). En este sentido, tenemos la información de referencia para iniciar la etapa de optimización presentada en la figura 6.1. El algoritmo de detección utilizado es el desarrollado por N. Burrus et al [33].

Los datos fijos residen en memoria RAM durante todo el tiempo de vida del proceso. Tan sólo es necesario la incorporación de la imagen real y pose inicial para arrancar una nueva ejecución. En este sentido, se consigue aumentar la eficiencia en términos de tiempo y coste computacional.

6.3.2. Inicialización del algoritmo

Los algoritmos de extracción de puntos característicos estudiados en el capítulo 5 trabajan con imágenes en escala de grises. En este sentido, se precisa de una etapa inicial que normalice las imágenes modelo y real que obtenemos de la cámara. Una vez adaptadas, se procede a extraer los puntos de interés.

Harris, Harris-Laplace, SUSAN, SIFT o SURF son algunos de los algoritmos de extracción de puntos característicos más destacados. En concreto, se desarrolla el algoritmo SURF, si bien cabe remarcar que es una

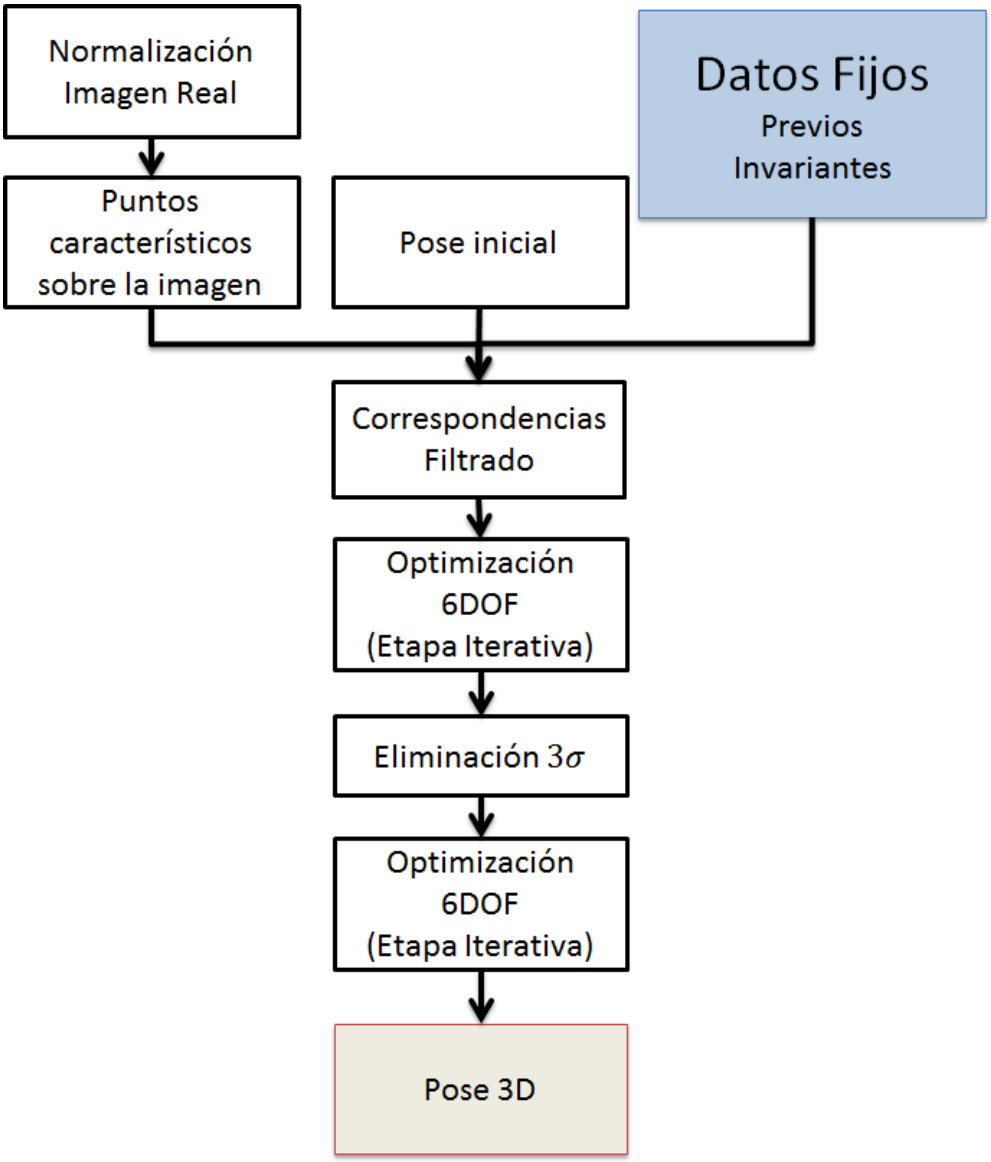


Figura 6.1: Descripción general del algoritmo.

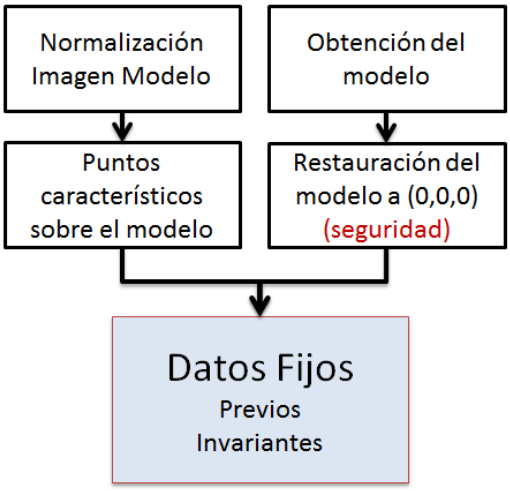


Figura 6.2: Datos fijos del algoritmo.

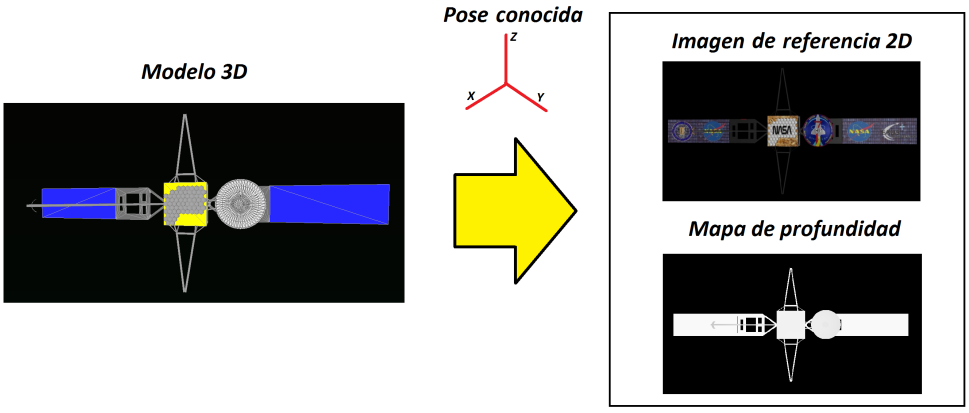


Figura 6.3: Imagen 2D y mapa de profundidad de referencia.

etapa intercambiable y reajustable al tipo de extractor que se necesite en función al contexto aplicativo.

Por último, se realiza la etapa de correspondencias entre los puntos característicos de la imagen de referencia y la imagen real. Para lograr una mayor precisión y reducir el número de falsas asociaciones, se introduce una etapa de filtrado que controla los parámetros angulares, escalares y de pose relativos al descriptor asociado a cada punto de interés.

6.3.3. Etapas iterativas

El algoritmo consta de tres etapas iterativas: dos de optimización y una etapa intercalada de eliminación de *outliers* (figura 6.1).

La función optimizadora se basa en el algoritmo de Levenberg-Marquardt. La decisión más importante es el determinación de la función de coste o error, así como el método seguido para calcularlo. En la sección 6.5.2.1 se analizan en detalle las decisiones implementadas.

Una vez superada la primera fase de optimización se procede a eliminar las correspondencias corruptas. Por último, se realiza una segunda etapa de optimización a partir de los parámetros de pose estimados en la primera fase.

6.3.4. Datos de salida

El resultado deseado es el que determina la pose 3D correcta del objeto modelo en la imagen real. Como queda reflejado en la sección 4.2.2, el resultado se representa mediante la siguiente matriz de dimensiones 4×4:

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El objetivo principal es calcular los parámetros de rotación dados por la matriz de coeficientes r_{ij} y el vector de translación $t = (t_1, t_2, t_3)^T$. Si

la estimación de dichos parámetros es correcta, es posible reconstruir el modelo 3D sobre la imagen real y comprobar visualmente el resultado.

6.3.5. Datos ejemplo

En esta sección se presenta el modelo $3D$ utilizado para realizar una explicación visual de los conceptos teóricos.

El modelo elegido es el módulo lunar (LEM) Apolo, primera nave diseñada para volar en el vacío sin ningún tipo de capacidad aerodinámica. A continuación, se presentan los datos fijos, previos a la ejecución del algoritmo.

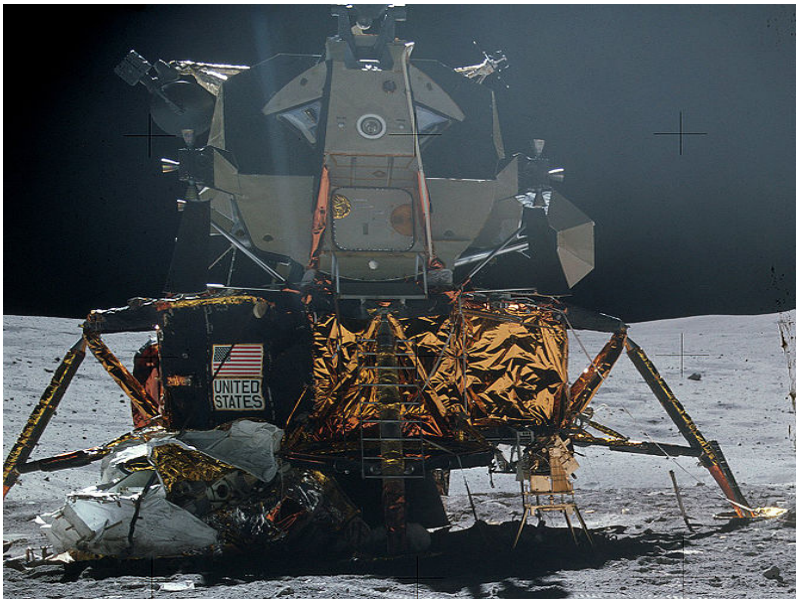
- Modelo 3D. En la figura 6.4 se expone el modelo 3D de LEM.

- Imagen 2D de referencia (figura 6.5). Esta imagen será utilizada como plantilla para realizar el *matching* con la imagen real que obtenemos directamente de la cámara. Los parámetros de pose del objeto son conocidos. En este sentido, se establece un punto de referencia para determinar la pose del objeto en la imagen real.

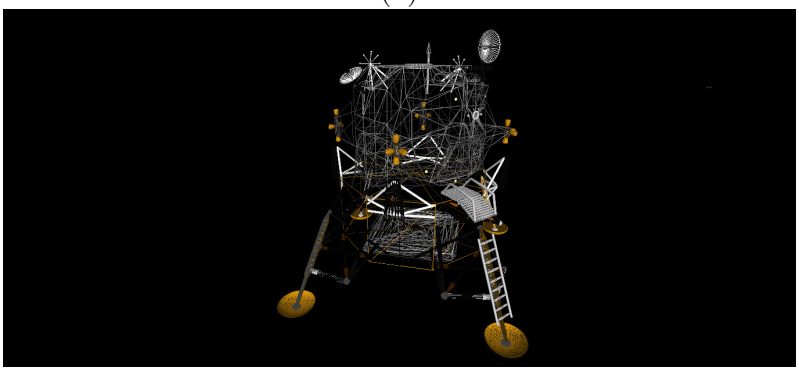
6.4. Puntos de interés

En esta sección se analizan las técnicas de extracción de puntos de interés más destacadas con el objetivo de determinar el algoritmo que mejor se ajuste al contexto aplicativo del proyecto. Independientemente del algoritmo elegido, cabe remarcar, que éste es un módulo intercambiable, es decir, ***la utilización de otro tipo de algoritmo en esta sección es válida y no modifica en ningún caso la estructura o morfología del método propuesto.***

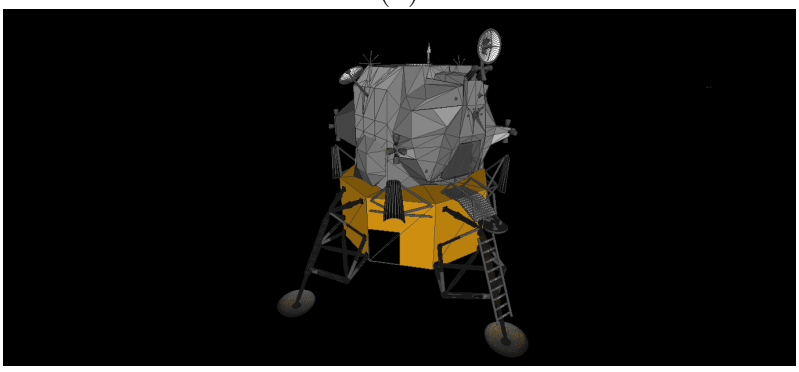
Se han realizado numerosos estudios comparativos. Entre ellos destaca el realizado por Johannes Bauer, Niko Sünderhauf y Peter Protzel de la universidad de Chemnitz en el año 2007 [22]. A modo de recordatorio (ver sección 5.7), el estudio ofrecía los siguientes resultados:



(a)



(b)



(c)

Figura 6.4: Modelo 3D de módulo lunar Apolo. En (a) se presenta una imagen real de LEM. En (b) y (c) se presenta la estructura de vértices y caras del modelo respectívamente.

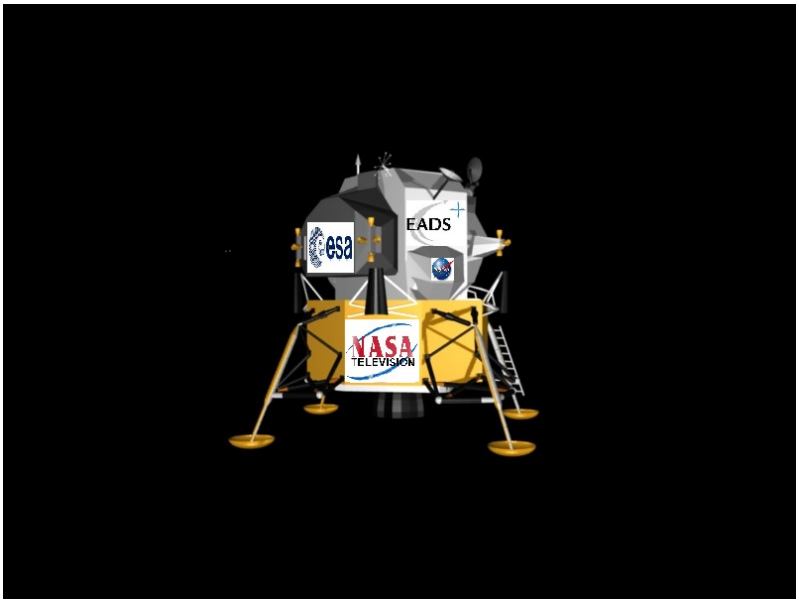


Figura 6.5: Imagen 2D del modelo LEM.

- Detectores como SUSAN, Harris o Harris-Laplace presentan resultados peores en todos los sentidos a SIFT o SURF.

- El número de punto de interés extraídos o detectados es mayor en SIFT que en SURF. La calidad de las asociaciones que realiza es similar, ligeramente superior en SIFT.Ambos algoritmos presentan buenos resultados ante transformaciones espaciales y ruido con pequeños porcentajes de error. Existe un determinado umbral de iluminación a partir del cual no se detecta prácticamente ninguna asociación. Para valores inferiores a dicho umbral el par de algoritmos es robusto ante cambios de luminosidad, ofreciendo resultados constantes. Por último, el tiempo de cómputo es superior en SIFT que en SURF.

A la vista de las conclusiones que se presentan, detectores clásicos como Harris quedan descartados ya que han sido complétamente superados por algoritmos más actuales como SIFT o SURF. Los resultados que ofrecen estos últimos son prácticamente equivalentes. Es cierto que SIFT realiza

correspondencias con porcentajes de error lévemente menor, pero el tiempo que necesita es mayor. Éste último factor nos conduce a elegir SURF como algoritmo de detección de puntos característicos.

Para más información en lo referente a la comparación de puntos de interés, ver sección 5.7, donde además se incluyen gráficas cualitativas que caracterizan los resultados particulares a cada algoritmo en distintos escenarios de aplicación.

6.4.1. SURF

En esta sección se realiza una aproximación práctica al algoritmo seleccionado para la fase de detección de puntos característicos, SURF. El objetivo es situar el algoritmo en el contexto de aplicación del proyecto para así poder diseñar el resto de fases que lo constituyen.

En la figura 6.4.1 se presenta el efecto de aplicar SURF a un par de imagenes en la que se sitúa el objeto bajo análisis. En las figuras 6.4.1.a y 6.4.1.c se muestra el par de imagenes reales, captura de la cámara. En las figuras homónimas 6.4.1.b y 6.4.1.d se presentan las imagenes resultado de aplicar el algoritmo de extracción de puntos de interés.

La pose del objeto es idéntica en el par de tomas, si bien el escenario varía. Aunque es pronto para analizar la robustez del algoritmo, podemos observar como los puntos característicos son idénticos en la estructura interna del objeto en ambas imágenes. Como era de esperar, en las regiones próximas a los bordes del modelo se producen diferencias, dado el cambio de escenario. En este sentido, se establece una condición fundamental para la aplicación de SURF como algoritmo de detección de puntos característicos: El objeto ha de tener textura para que pueda ser reconocido independientemente del escenario en el que se encuentre.

En la sección 5.6.3 se realizaba un estudio sobre la estructura y forma del descriptor SURF. Además de la información propia del descriptor, el algoritmo implementado aporta información sobre el entorno que ha considerado en la construcción de dicho descriptor (tamaño de la región bajo estudio, dirección predominante y gradiente). En la figura 6.7 podemos observar de manera más concisa el efecto de aplicación del algoritmo. Co-

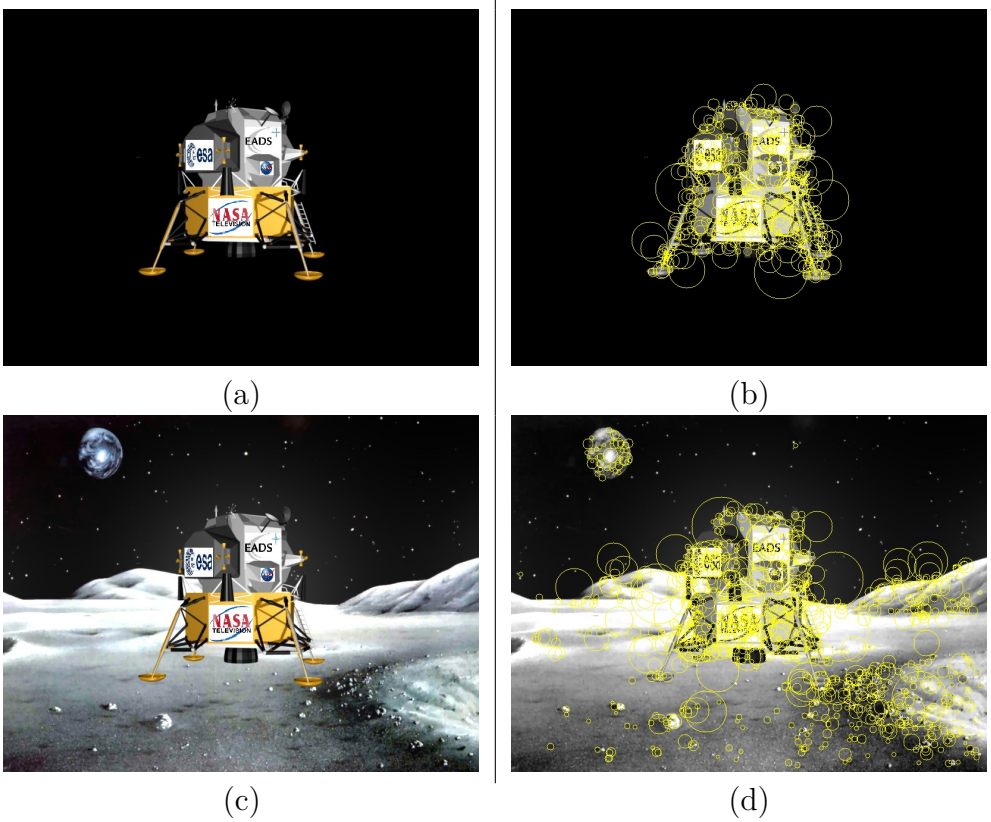


Figura 6.6: Características SURF sobre imagen. En (a) y (c) se muestra el par de imágenes que toma la cámara del objeto LEM en dos escenarios diferenciados. En (b) y (d) es ilustra la extracción de puntos característicos en cada una de ellas.

Figura 6.7: Características SURF sobre imagen.

mo se puede observar, sobre cada punto SURF se sitúa una circunferencia que describe el sector bajo análisis. Asociado a dicho sector se almacena la información de gradiente y dirección principal. Esta información se usa en la siguiente etapa para filtrar las falsas correspondencias y reducir el error total.

Para más información acerca de SURF, consultar sección 5.6 donde se realiza un análisis completo del algoritmo.

6.4.2. Etapa de correspondencias

El objetivo fundamental del algoritmo es el refinamiento de pose 3D. De esta afirmación se desprende una idea fundamental: si estamos refinando una pose, entonces el objeto que se sitúa en la imagen real tendrá una pose cercana a la de la imagen modelo 2D. Entonces, los puntos característicos de la imagen modelo de coordenadas (x, y) estarán próximos a los de la imagen modelo (x', y'). Además, parámetros internos SURF como el radio o sección característica, así como la orientación, gradiente o dirección han de ser parecidos (entorno prácticamente idéntico). En este sentido, aparece una etapa intermedia de filtrado.

6.4.2.1. Etapa de filtrado

Se definen los siguientes filtros:

- *Filtro espacial*. El punto característico p de coordenadas (x, y) de la imagen A sólo puede asociarse con el punto p' de la imagen B si las coordenadas de dicho punto están en un círculo de radio r centrado en el punto (x, y) de B y cumple el resto de restricciones. Otras formas geométricas son válidas, caso del rectángulo, elipse, etc. La elección depende del contexto aplicativo.

- *Entorno o sector característico*. En este caso se compara la información del entorno asociada a cada punto característico. El punto característico p de la imagen A sólo puede asociarse con el punto p' de la imagen B si:

$$0{,}8r_{pA} < r_{p'B} < 1{,}2r_{p'A}$$

  donde r_{pA} y $r_{p'B}$ representan los radios asociados al punto p de la imagen A y al punto p' de la imagen B.

- *Vector de orientación*: Cada punto SURF contiene información sobre la dirección predominante del sector característico. En este sentido, el punto característico p de la imagen A sólo puede asociarse con el punto p' de la imagen B si:

$$dir_{pA} - 20^{\text{o}} < dir_{p'B} < dir_{pA} + 20^{\text{o}}$$

  donde dir_{pA} y $dir_{p'B}$ representan, en grados, las direcciones predominantes asociadas a los puntos característicos p y p' respectivamente. Por lo tanto, sólo se pueden asociar si sus direcciones están en un margen de 20^{o} de diferencia en valor absoluto.

- *Profundidad*. En el caso de tener información sobre la profundidad de la imagen, éste tipo de filtrado es de gran ayuda. De manera similar a lo explicado en los apartados anteriores, se dispone un rango de valores de profundidad asociados a cada punto de interés.

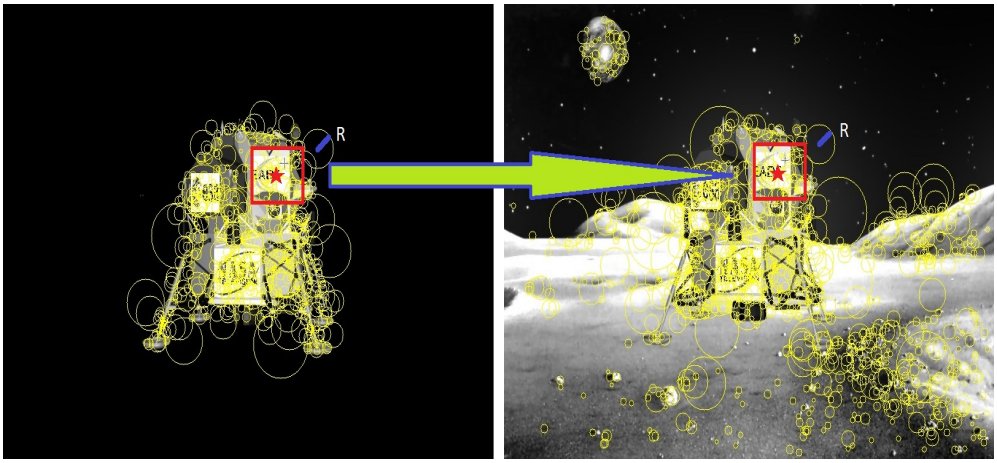


Figura 6.8: Filtrado visual sobre imagen modelo y real.

En este sentido, el punto característico p de la imagen A sólo puede asociarse con el punto p' de la imagen B si:

$$0{,}8pr_{pA} < pr_{p'B} < 1{,}2pr_{pA}$$

donde pr_{pA} y $pr_{p'B}$ representan las profundidades asociadas al punto p de la imagen A y al punto p' de la imagen B respectivamente.

El resultado inmediato es una optimización del tiempo de ejecución del algoritmo, ya que si no cumple una de las restricciones, se elimina directamente del conjunto de candidatos y se procede a evaluar el siguiente. En la figura 6.8 se expone visualmente el filtrado propuesto, en el caso particular de utilizar un filtro espacial tipo rectangular. Además, representa un primer contexto de aplicación en el que hay un par de imágenes, con diferentes escenarios y por tanto diferente número y congestión de puntos SURF. La probabilidad de que un punto de la imagen de la izquierda (fondo homogéneo) coincidiera con un punto del entorno, escenario de la figura de la derecha, no es despreciable.

Existen múltiples razones que motivan la implementación de esta etapa de filtrado en el contexto de refinamiento de pose 3D. Las principales son:

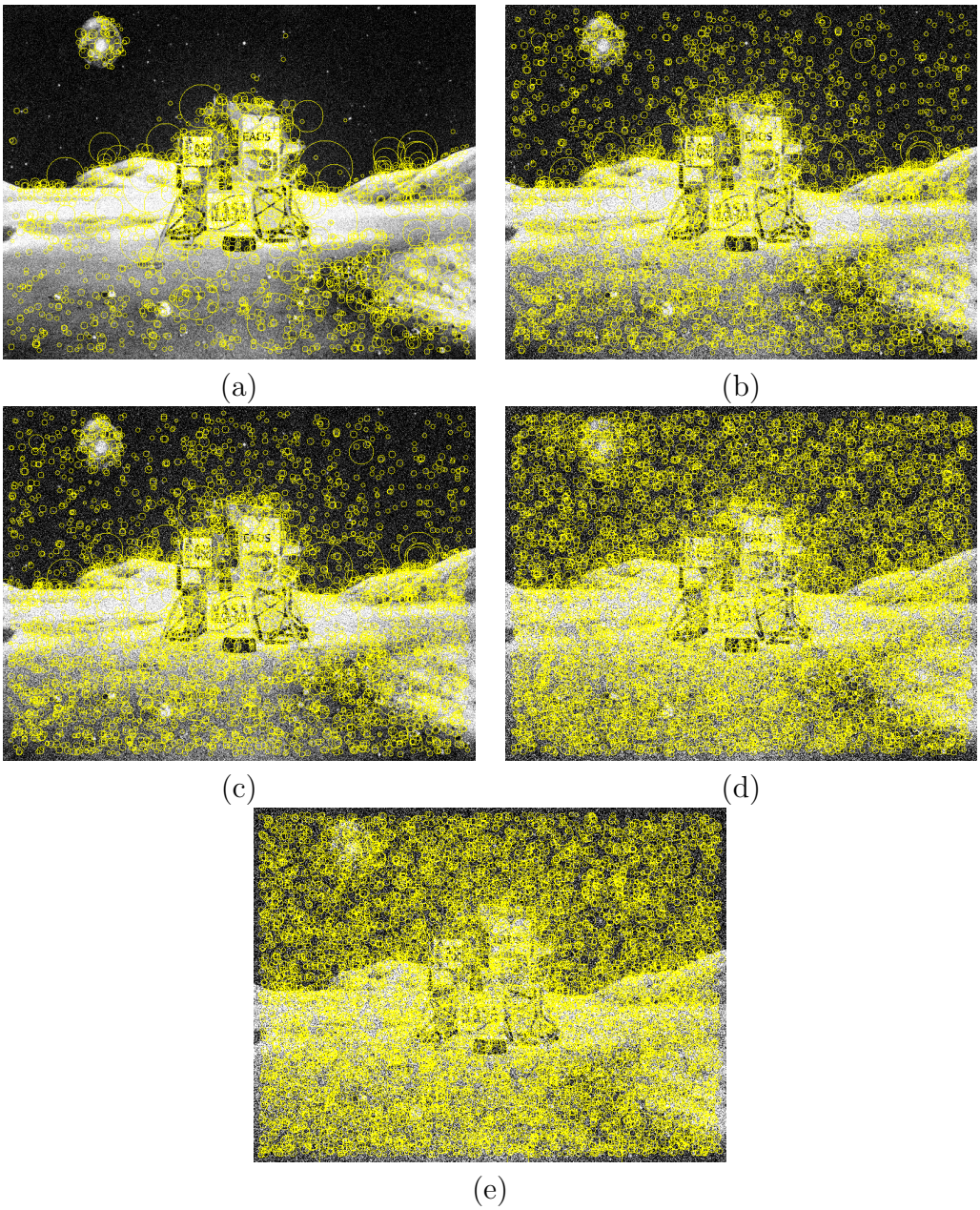
- Ruido en el proceso de formación de la imagen.
- Visualización parcial del objeto.
- Tipo de escenario en el que se encuentra el objeto.

Ruido El ejemplo de la figura 6.9 presenta los efectos que tiene el ruido sobre el número de puntos característicos extraídos. A medida que el nivel de ruido aumenta, el número de puntos de interés resultante crece. En este sentido, la probabilidad de correspondencia entre un par de puntos alejados o erróneos se incrementa en la misma proporción que dicho nivel. Teniendo en cuenta el contexto aplicativo en el que nos encontramos, sabemos que el objeto debe estar en una pose próxima a la del modelo de referencia, y por lo tanto la aplicación del filtrado supone una ayuda fundamental y necesaria para minimizar el número de falsas correspondencias. Si limitamos el sector de búsqueda, así como el resto de parámetros, la probabilidad de error se reduce de forma significativa.

En la figura 6.10 se presenta el efecto de aplicar o no la etapa de filtrado sobre el conjunto de imagenes ruidosas. En la columna de la izquierda se observa como la no utilización de filtrado conduce a resultados erróneos en la mayoría de los casos, si bien es más pronunciado a medida que se acentúa el efecto. El número de asociaciones permanece alto ya que se generan múltiples puntos con características semejantes a las del modelo original sin ruido. Por otro lado, si aplicamos la etapa de filtrado propuesta, ver columna de la derecha, se observa un decremento cuantitativo del número de asociaciones erróneas. Sin embargo, el número de correspondencias disminuye debido al cambio de contexto.

Se presentan dos alternativas:

- Definir filtros muy restrictivos acordes al contexto aplicativo, con el objetivo de obtener un número mínimo de falsas asociaciones. Hay que tener en cuenta que el número de correspondencias resultado es bajo y en general se pueden producir pérdidas de asociaciones correctas. Si además el objeto tiene poca textura, podemos estar ante



(a) (b) (c) (d) (e)

Figura 6.9: Ruido gaussiano progresivo sobre imagen real. La secuencia ilustra la relación entre el nivel de ruido y el número de puntos de interés extraídos.

problemas de convergencia cuando la escena en el que se encuentra el objeto varía.

- Definir filtros poco restrictivos y aplicar algoritmos como RANSAC para eliminar los posibles outliers presentes en la muestra. En este contexto, el número de asociaciones crece, al igual que el porcentaje de falsas correspondencias.

En general, el uso de filtros muy restrictivos se desaconseja y se opta por la implementación de filtros menos restrictivos en conjunción con algoritmos como RANSAC. Más adelante se presenta una comparación entre ambas alternativas.

Visualización parcial del objeto Un casó típico de aplicación surge cuando un objeto "no deseado" aparece en la imagen, ocultando parte de la superficie del objeto bajo análisis y del cual queremos determinar su pose 3D. La figura 6.11 presenta la interferencia de un astronauta en el campo de visión del módulo lunar LEM. Como resultado, sólo se aprecia parte de su estructura. Se introducen nuevos puntos SURF pertenecientes al objeto interferente, si bien la probabilidad de asociaciones en dicha región es baja gracias al aporte de la fase de filtrado y la robustez de SURF. En la figura 6.11 podemos comprobar como no se realiza asociación alguna en la región de interferencia a pesar de la incorporación de nuevos puntos característicos.

Tipo de escenario El medio en el que se encuentra el objeto representa un factor clave en el proceso de identificación y detección. En la figura 6.4.1 se introducía un escenario moderado que presentaba un número no muy elevado de puntos característicos. El contexto ideal es aquel que no presenta ningún punto de interés y sólo existe información del objeto deseado (escenario homogéneo). Sin embargo, como se puede observar en la figura 6.12, existen multitud de escenarios cada uno con diferente nivel de caracterización. A medida que el escenario es más característico, la detección empeora, produciéndose un efecto similar al del ruido, en el que aparece un mayor número de puntos de interés semejantes a los del

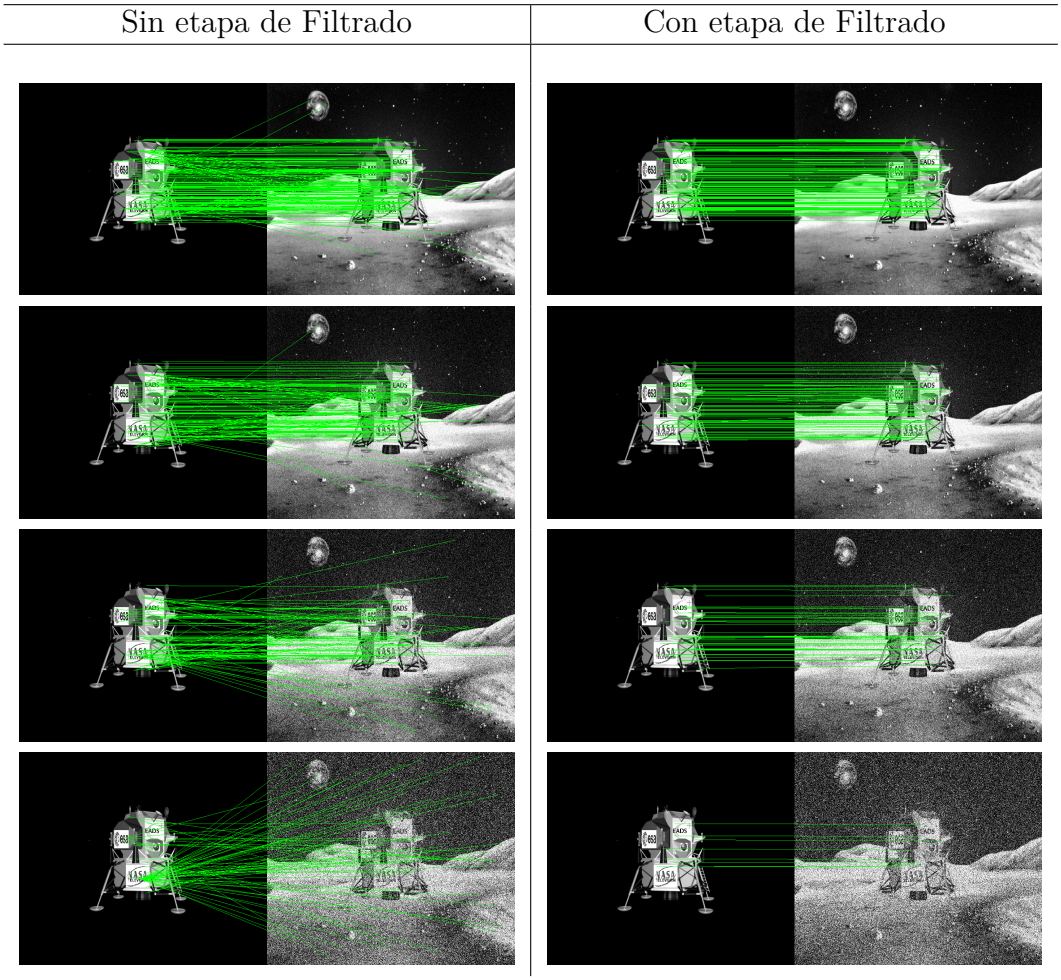
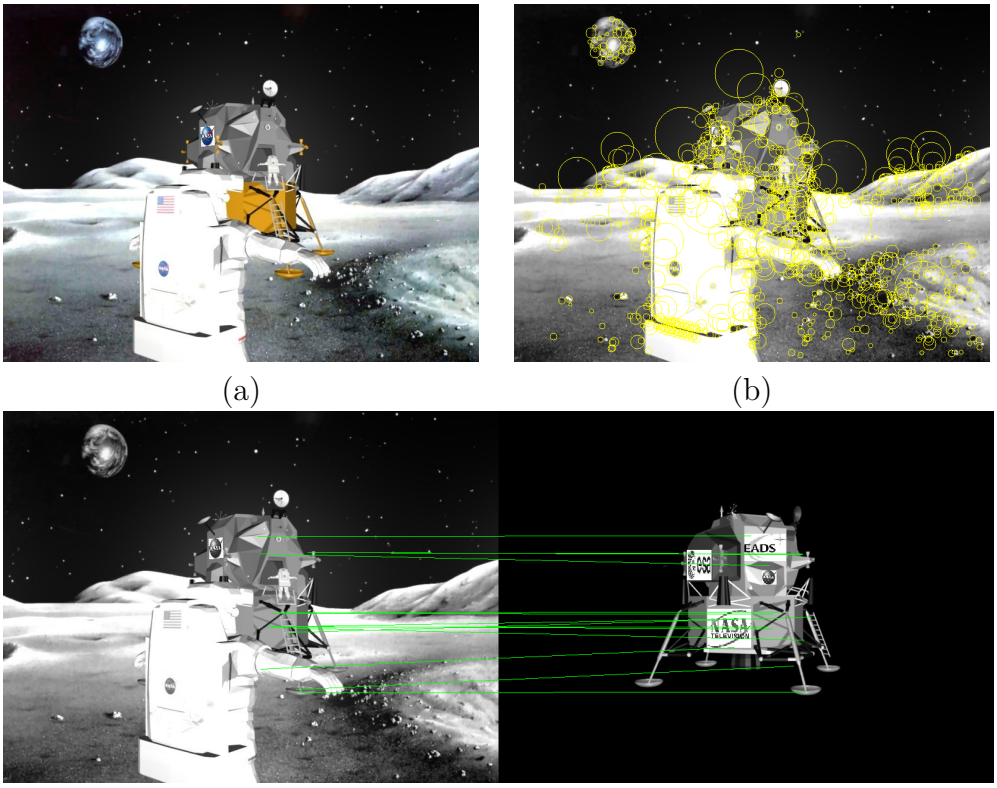


Figura 6.10: Correspondencias finales en presencia de ruido. En la figura se presentan los resultados de la etapa de correspondencias con/sin etapa de filtrado inicial.



(a) (b)

(c)

Figura 6.11: Visualización parcial de módulo lunar LEM. En (a) se presenta la imagen que se obtiene directamente de la cámara. En (b) se presenta la aplicación de SURF sobre la imagen normalizada. La figura (c) ilustra la fase de correspondencias.

objeto bajo estudio. La etapa de filtrado reduce el espacio de búsqueda minimizando dichos efectos.

6.4.2.2. Comparación de descriptores

Una vez realizada la etapa de filtrado, se procede con la comparación de descriptores candidatos restantes. Se establecen diferentes niveles de similitud, representados mediante la variable $k > 0$.

En la figura 6.13 se presenta el conjunto de correspondencias finales para el mismo par de imagenes con variación del parámetro k. Como

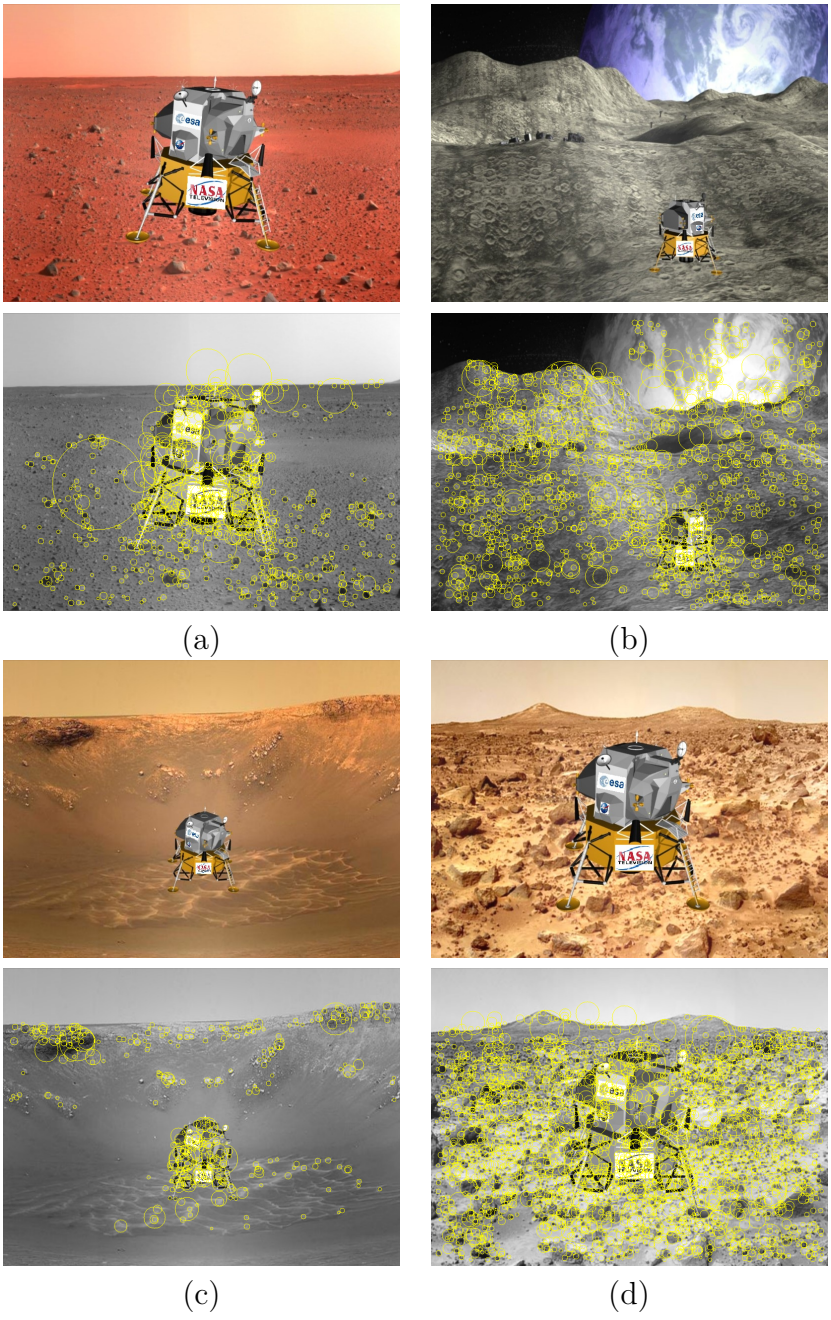
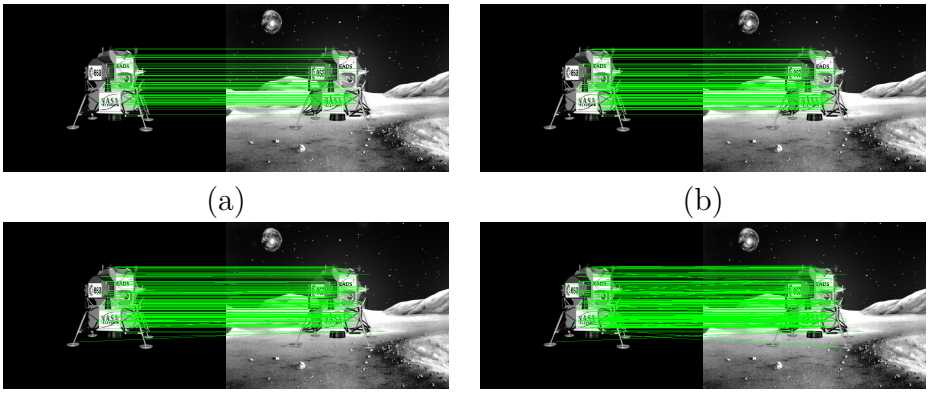


Figura 6.12: Tipos de escenario. En la secuencia de imágenes se observa al objeto LEM en diversos tipos de escenario. Cada uno de ellos presenta un nivel diferente de puntos característicos, lo que determina de forma directa la capacidad de detección del algoritmo. Los escenarios (b) y (d) representan entornos altamente característicos, si bien los escenarios (a) y (c) representan entornos más suaves donde la detección es más fácil.



(a) (b)

(c) (d)

Figura 6.13: Ajuste del parámetro de correspondencia. La secuencia ilustra la variación de correspondencias finales en función a la variación del parámetro k. En (a) se utiliza un valor $k = 0{,}2$. Éste valor se incrementa proporcionalmente hasta concluir en (d) con valor $k = 0{,}8$. La consecuencia es el aumento en el número de correspondencias y outliers resultante.

se puede comprobar, a medida que disminuimos el grado de similitud k, el número de correspondencias aumenta, ya que estamos siendo menos restrictivos. Sin embargo, también se produce un aumento del número de falsas correspondencias, factor en cualquier caso no deseado. Al igual que en la etapa anterior, existen dos alternativas:

- Utilizar valores de correspondencia altos. En este sentido, la probabilidad de obtener asociaciones correctas es elevada. Sin embargo, el número de asociaciones es bajo.

- Utilizar valores de correspondencia bajos. En este caso el número de asociaciones es elevado pero aparecen falsas asociaciones. Es en este contexto donde tiene sentido la aplicación de algoritmos como RANSAC que eliminan las falsas correspondencias, utilizando únicamente las asociaciones válidas para converger a la solución adecuada.

En secciones posteriores se analizan los efectos de ambas alternativas, si bien, cabe adelantar que en la mayoría de las ocasiones es necesaria la

aplicación de la segunda opción, ya que aparecen problemas de convergencia derivados del reducido número de puntos que se determinan.

La figura 6.14 presenta las consecuencias de utilizar un valor de k *demasiado restrictivo*. En la figura 6.14.a se presenta la correspondencia entre un par de imagenes con el mismo objeto y pose. El número de correspondencias es elevado ya que las imágenes son idénticas. En la figura 6.14.b se introduce una rotación de 5^{o} respecto al eje Z del objeto. El número de correspondencias se ha reducido sustancialmente, si bien el cambio de orientación y pose es mínimo. Además, la mayoría de las asociaciones que se establecen se sitúan en los bordes del objeto, con lo que un cambio de escenario terminaría de reducir el número de correspondencias (ver figura 6.14.c), produciéndose un problema grave de convergencia.

6.5. Estimación de pose

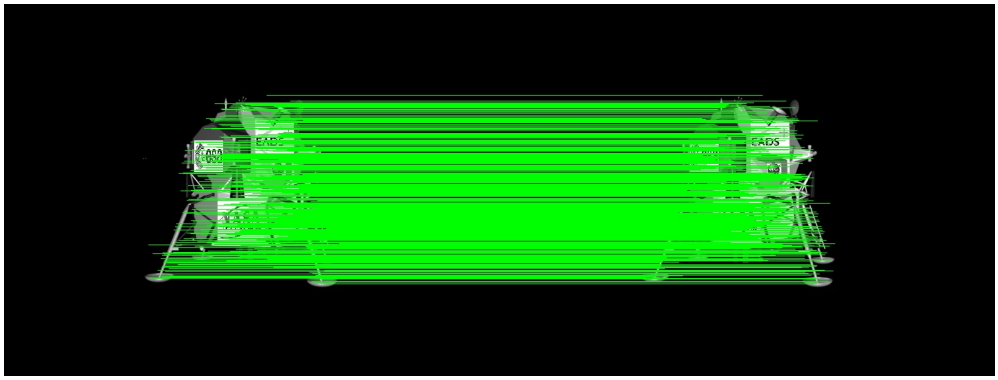
En esta sección se presentan los algoritmos implementados para solucionar el problema de estimación de pose $3D$. Se plantean dos alternativas: POSIT y una solución de optimización basada en mínimos cuadrados, posteriormente mejorada mediante la incorporación de RANSAC. En definitiva, *se desarrollan* ***tres*** *algoritmos* cuyas prestaciones y características se analizarán en el capítulo 7.

6.5.1. POSIT

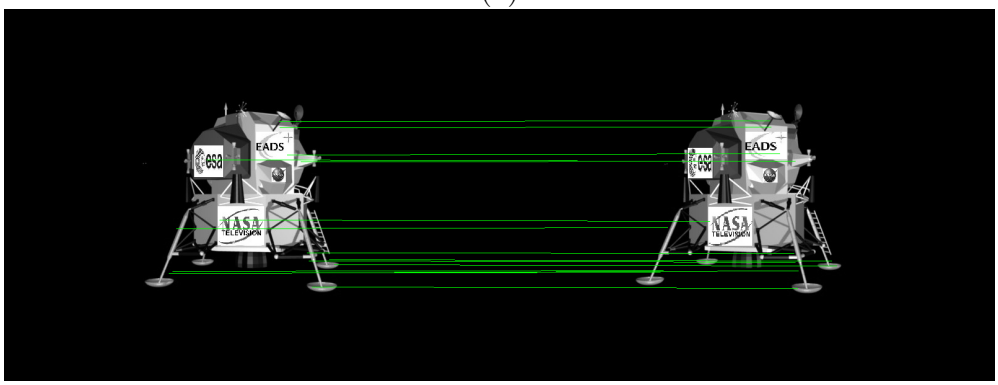
El algoritmo POSIT es uno de los métodos más populares en el campo de la estimación de parámetros de pose 3D de un objeto. En este sentido, es importante realizar su implementación, tal como se describe en la sección 4.6 para poder realizar una comparación cualitativa del algoritmo que se propone.

6.5.2. Optimización vía mínimos cuadrados

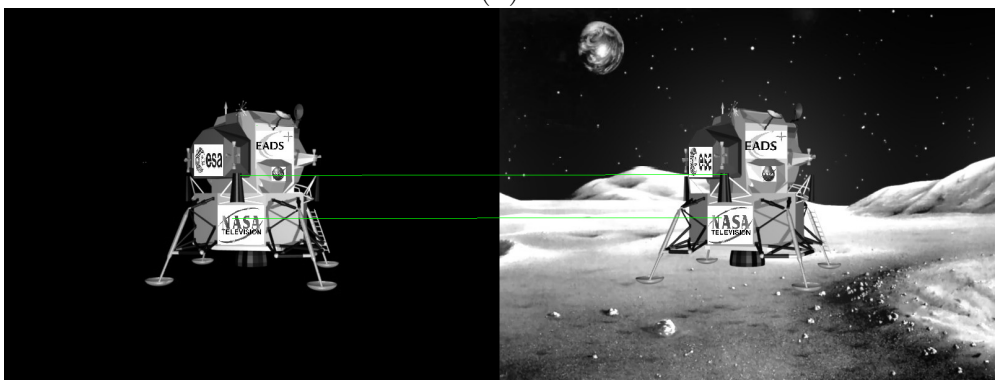
El contexto y aplicación en el que nos encontramos caracteriza de manera directa la estructura y forma del algoritmo. Por un lado, el hecho



(a)



(b)



(c)

Figura 6.14: Relación entre el parámetro de correspondencia y la convergencia del algoritmo. En (a) se ilustra la etapa de correspondencias para un par de imágenes idéntico utilizando un valor de k restrictivo. En (b) se ha producido un cambio de 5º respecto a la original utilizando el mismo valor de k. En (c) se ilustra la transformación (b) con un cambio de escenario.

de partir de una pose cercana a la real, refinamiento de pose, nos lleva a emprender la búsqueda de un **mínimo local** en dicho entorno. Existen, como queda reflejado en el capítulo 4, varios métodos de tipo algebraico que buscan el mínimo global. Dichos algoritmos sólo convergen si los datos son correctos, es decir, todas las asociaciones o correspondencias se han realizado con éxito. En un contexto general, esta circunstancia no es posible, y además, la búsqueda del mínimo global no es necesaria ya que partimos de una inicialización próxima a la deseada.

El algoritmo de optimización consta de las siguientes etapas (figura 6.15):

- Cálculo del error parcial (etapa i). Se detalla en la sección 6.5.2.1.
- Optimización y determinación de los nuevos parámetros de pose. Se detalla en la sección 6.5.2.1. Esta etapa requiere la información de error calculado en el paso anterior.

Si el error inicial es aceptable (comprobación con parámetros de pose estimados por el método de detección o de la pose anterior, tracking), el método no se ejecuta y termina. La razón principal de esta comprobación es la eficiencia en cuanto a tiempo y coste computacional. Si por el contrario, el objeto está alejado, se procede con el método iterativo consistente en las dos etapas mencionadas en el párrafo anterior: estimación de parámetros de pose y cálculo de error. Por último, se comprueba si la estimación es válida.

A continuación se detalla el cálculo del error, así como la optimización por mínimos cuadrados implementada mediante Levenberg-Marquardt.

6.5.2.1. Cálculo del error

En un contexto de optimización, la definición del error es fundamental para entender el proceso completo.

Una de las principales ventajas de la implementación de algoritmos iterativos es la posibilidad de incluir, de forma sencilla, múltiples parámetros en el cálculo del error. Las mejoras que se producen son relevantes, ya que si el volumen de información que tenemos de la escena es mayor,

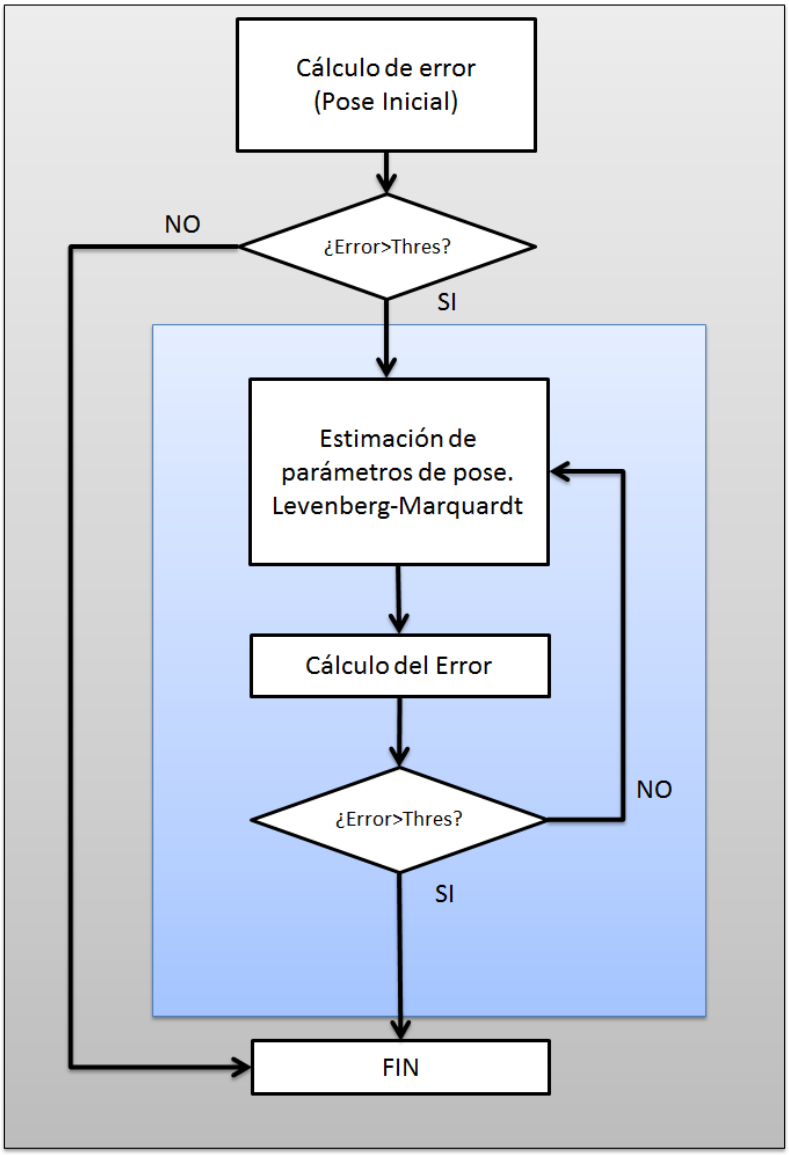


Figura 6.15: Descripción del algoritmo de estimación de pose.

la probabilidad de realizar una detección correcta aumenta. ***El método propuesto admite dos alternativas que derivan de la utilización o no de la información de profundidad.***

1ª Alternativa: Sin información de profundidad

En este caso *no se dispone de información de profundidad sobre la imagen real*.

El procedimiento de cálculo es el siguiente: Se toman los puntos de interés resultado de la fase de correspondencias sobre la imagen modelo y se realiza la proyección $2D \rightarrow 3D$ de dichos puntos con los parámetros de *pose de referencia*. La transformación mencionada se realiza gracias a que disponemos de información de profundidad asociada a la imagen modelo 2D. A continuación, se realiza la proyección $3D \rightarrow 2D$ con la *pose estimada* sobre la imagen real y se calcula la distancia con los puntos característicos homónimos de dicha imagen.

En términos matemáticos, el error de la etapa j (proceso iterativo), se expresa como sigue:

$$E_j = \sum_{i=1}^{N} e_i$$

con

$$e_i = D\left(T^R_{3D\rightarrow 2D}\left(T^M_{2D\rightarrow 3D}\left(p_i^M\right)\right), p_i^R\right)$$

donde

1. e_i representa el error asociado al par i-ésimo de correspondencias.
2. p_i^M y p_i^R representan los puntos característicos $2D$ resultado de la fase de correspondencias sobre la imagen modelo y real respectivamente.
3. $T^M_{2D\rightarrow 3D}\left(p_i^M\right)$ representa la proyección $3D$, con parámetros de pose iniciales (modelo), de los puntos característicos $2D$ de la imagen de referencia. Esta transformación es posible gracias a que se dispone de la información de profundidad asociada a dicha toma.

4. $T^{R}_{3D\to 2D}\left(T^{M}_{2D\to 3D}\left(p_i^M\right)\right)$ representa la proyección $2D$, con parámetros de pose reales (estimados), de los puntos $3D$ del apartado anterior sobre la imagen real.

5. $D\left(f,g\right)$ representa la distancia euclídea entre f y g.

Mapa de asociaciones La figura 6.16 presenta el proceso completo de generación del mapa de asociaciones. El punto de partida es el par de imágenes real y modelo a las que se aplica el algoritmo de extracción de puntos característicos (figura 6.16.a). Se realiza la etapa de filtrado y correspondencias (figura 6.16.b). El mapa de asociaciones (figura 6.16.c) representa, en el plano bidimensional, los puntos de interés resultantes de la etapa de correspondencias.

Coordenadas 3D mapa modelo Las coordenadas $3D$ del mapa modelo se calculan a partir de la transformación $2D \to 3D$ de los puntos característicos patrón (mapa de puntos de interés sobre imagen modelo $2D$) con parámetros de pose de referencia, ver figura 6.17. La motivación es disponer un conjunto de coordenadas 3D al que podamos aplicar sucesivas transformaciones de pose, las estimadas mediante el algoritmo, que transladen dichos puntos al escenario de la imagen real. Si evaluamos el error que se produce con dichas estimaciones, podemos valorar la calidad de las mismas. Además, desde un punto de vista computacional, el conjunto de datos resultante de esta primera transformación es fijo y por lo tanto no requiere de cálculo adicional, almacenándose en memoria RAM.

A continuación, se procede con una etapa iterativa que consiste en proyectar dichas coordenadas con los parámetros estimados de pose resultado de la etapa anterior $i-1$. El resultado es un nuevo mapa de coordenadas 2D que servirá para determinar la calidad de la estimación. En la figura 6.5.2.1 se puede visualizar de manera gráfica el procedimiento explicado.

Mapa distancias final Una vez obtenido el mapa de puntos proyectados con la pose estimada, se procede a calcular el error como se observa

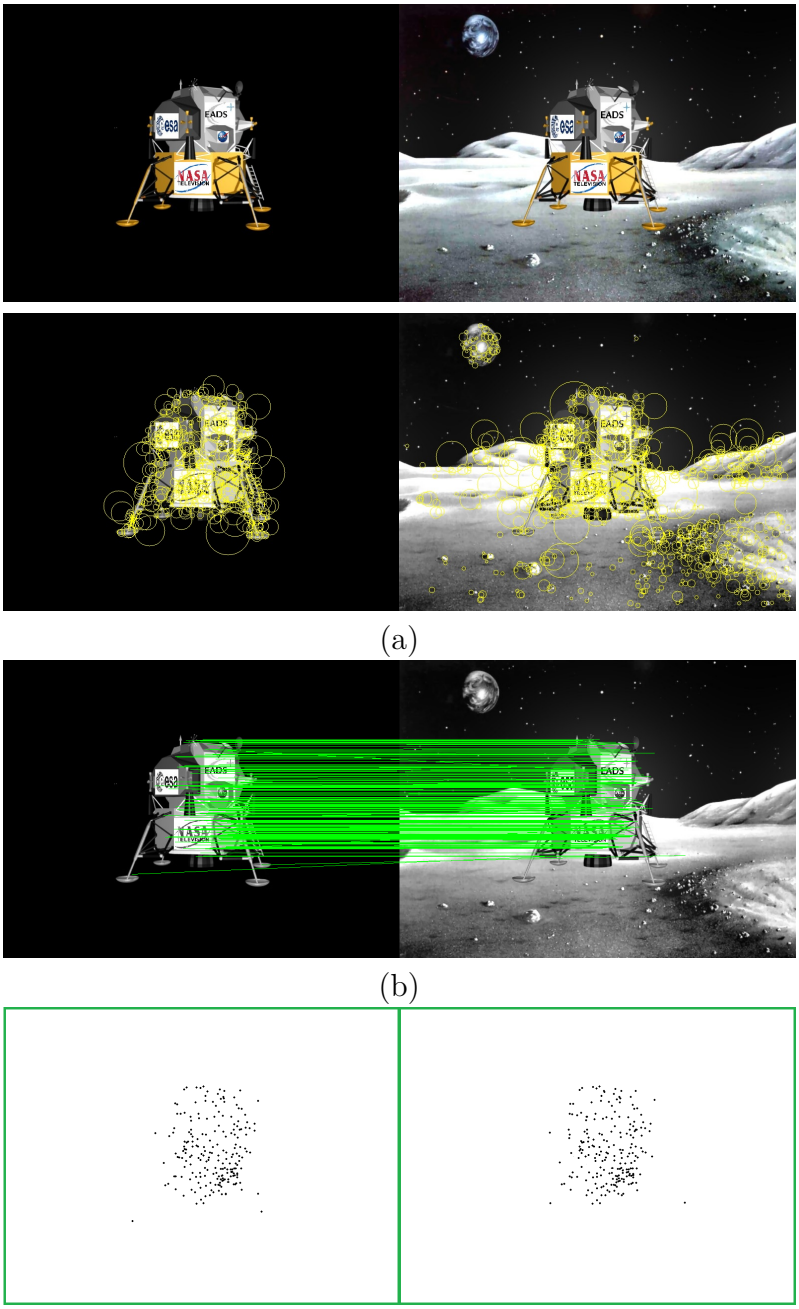


Figura 6.16: Mapas de puntos característicos. En (a) se ilustran los puntos interés en el par de imágenes real y modelo. En (b) se presentan los resultados de la etapa de correspondencias. Por último, se ilustra el mapa de puntos característicos del par de imágenes (c).

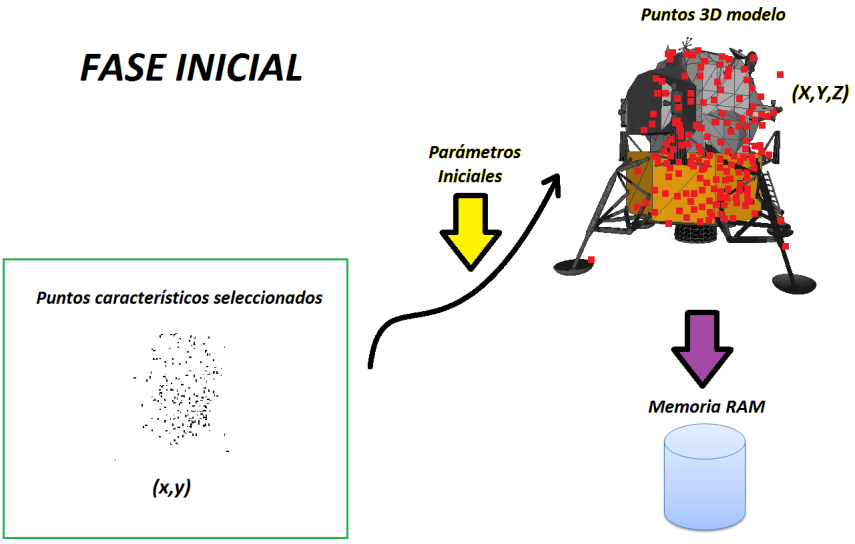


Figura 6.17: Coordenadas 3D modelo. En la figura se presenta el proceso de transformación geométrica inicial por el que se determinan las coordenadas 3D del mapa modelo. Una vez calculadas, se almacenan en memoria RAM.

en la figura 6.19. El mapa de distancias final se genera como la unión del par de mapas resultado de la fase iterativa y el de la imagen real obtenido al principio del algoritmo. El error total se calcula como la suma de todas las distancias, segmentos verdes sobre la imagen 6.19.

Una vez calculado el error, se realiza la optimización de la etapa i mediante Levenberg-Marquardt. El resultado es un conjunto de parámetros que reflejan la nueva estimación de pose.

2ª Alternativa: Con Información de profundidad

En este caso *tenemos información de profundidad sobre la imagen real*. Haciendo uso de los mapas de profundidad de la imagen modelo, podemos establecer *correspondencias entre puntos característicos* $3D$ y evaluar su error. El procedimiento de cálculo de error es similar al presentado en la primera alternativa. La diferencia radica en la segunda transformación

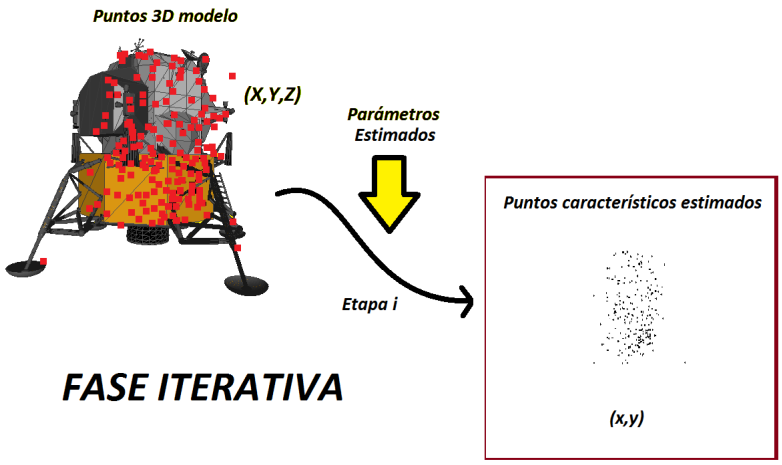


Figura 6.18: Etapa de transformación iterativa. Generación de mapa de puntos característicos estimados.

realizada (figura 6.5.2.1). Una vez tenemos los puntos $3D$ característicos, se realiza la proyección $3D \rightarrow 3D$ con la *pose estimada* y se calcula la distancia con los puntos de interés homónimos $3D$ de la imagen real, gracias a la información de profundidad que proporciona la cámara ***ToF***.

En este caso, el error asociado al par de correspondencias i-ésimo se expresa como:

$$e_i = D\left(T^R_{3D\rightarrow 3D}\left(T^M_{2D\rightarrow 3D}\left(p_i^M\right)\right), P_i^R\right)$$

donde:

1. P_i^R es el punto característico $3D$ resultado de la fase de correspondencias sobre la imagen real.

2. $T^R_{3D\rightarrow 3D}\left(T^M_{2D\rightarrow 3D}\left(p_i^M\right)\right)$ representa la proyección $3D$, con parámetros de pose real (estimada), de los puntos $3D$ característicos del modelo (proyección $2D \rightarrow 3D$ de los puntos de referencia $2D$ con parámetros de pose inicial) sobre la imagen $3D$ real.

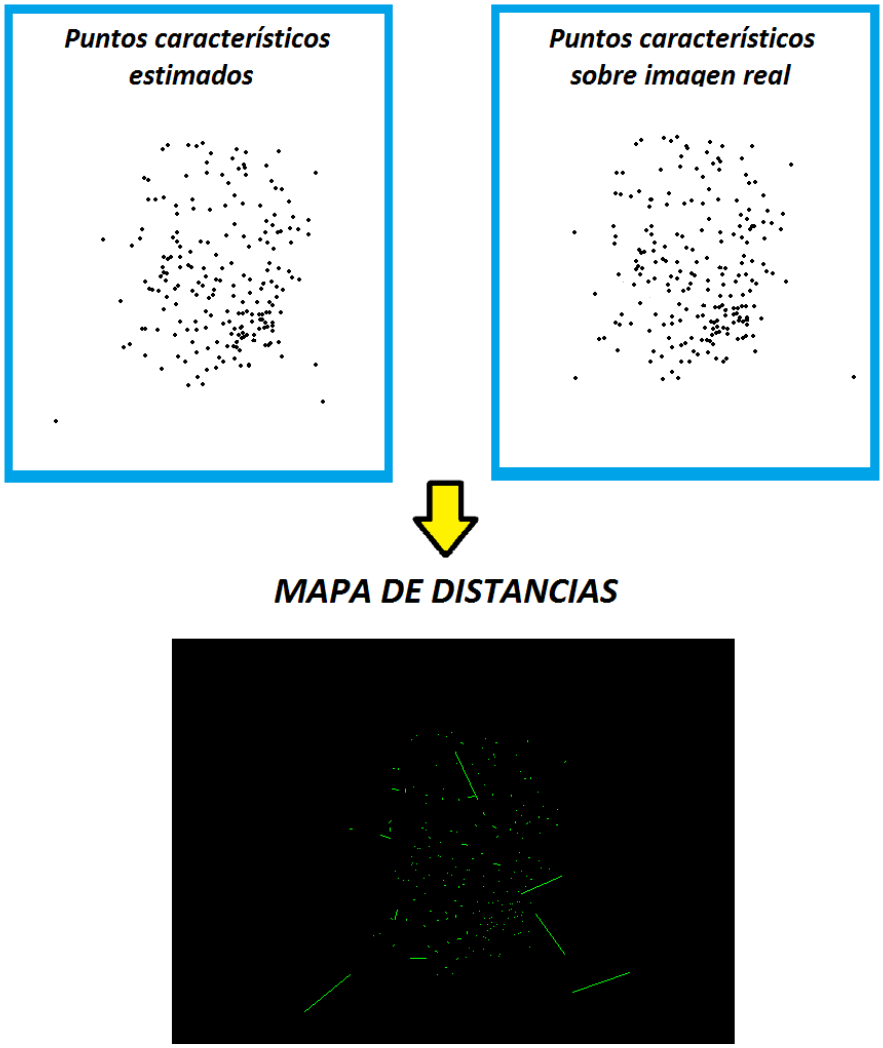
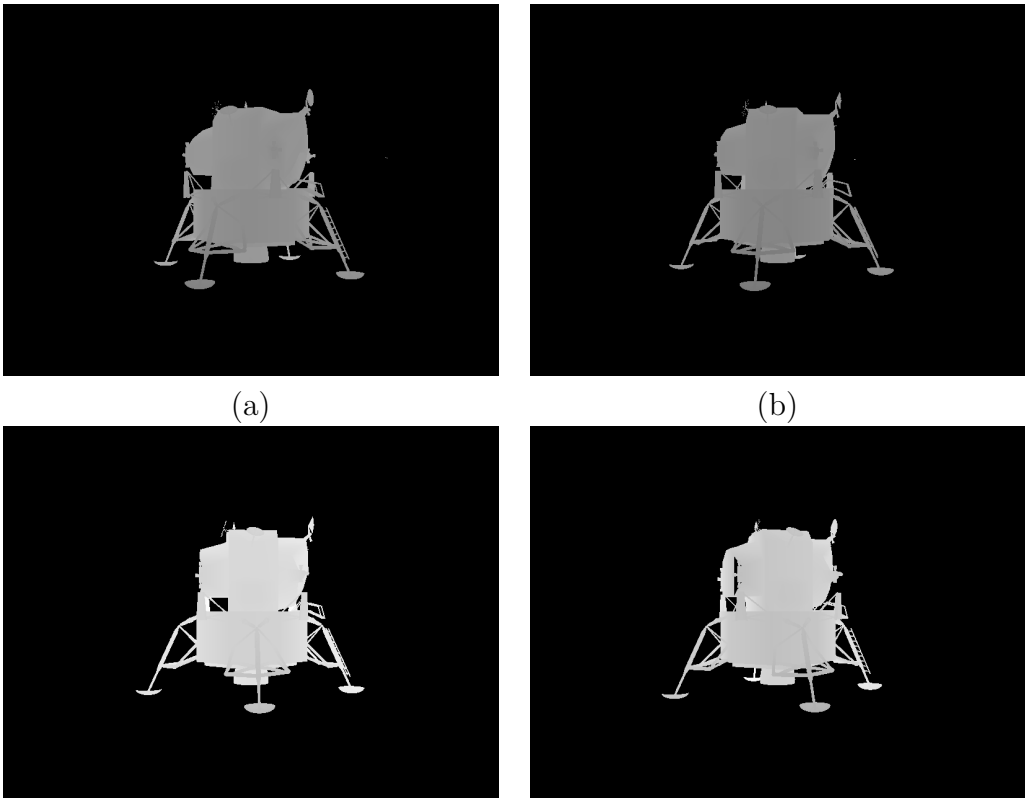


Figura 6.19: Mapa de distancias resultado. Los puntos característicos estimados son el resultado de la proyección de los puntos de interés 3D (fase de correspondencias) sobre la imagen real. Los puntos característicos sobre la imagen real son directamente los seleccionados tras la fase de asociaciones.



(a) (b)

(c) (d)

Figura 6.20: Mapas de profundidad.

En la figura 6.20 se presenta un ejemplo de mapas de profundidad asociados al modelo lunar LEM.

La respuesta del algoritmo se puede mejorar también en la fase de filtrado. *Si conocemos la profundidad del punto característico p en la imagen modelo, éste sólo podrá corresponder con los puntos p' de la imagen real que presenten una profundidad en un rango similar a la dada y además cumpla el resto de restricciones.*

Las cámaras ToF permiten realizar filtrado en tiempo real sobre distintos parámetros. En este sentido, si conocemos la profundidad aproximada de un objeto, podemos filtrar el rango de distancias en el cual queremos obtener la imagen. La figura 6.5.2.1 ilustra de manera gráfica este tipo de filtrado. Como se puede observar, se define un rango en el cual queremos

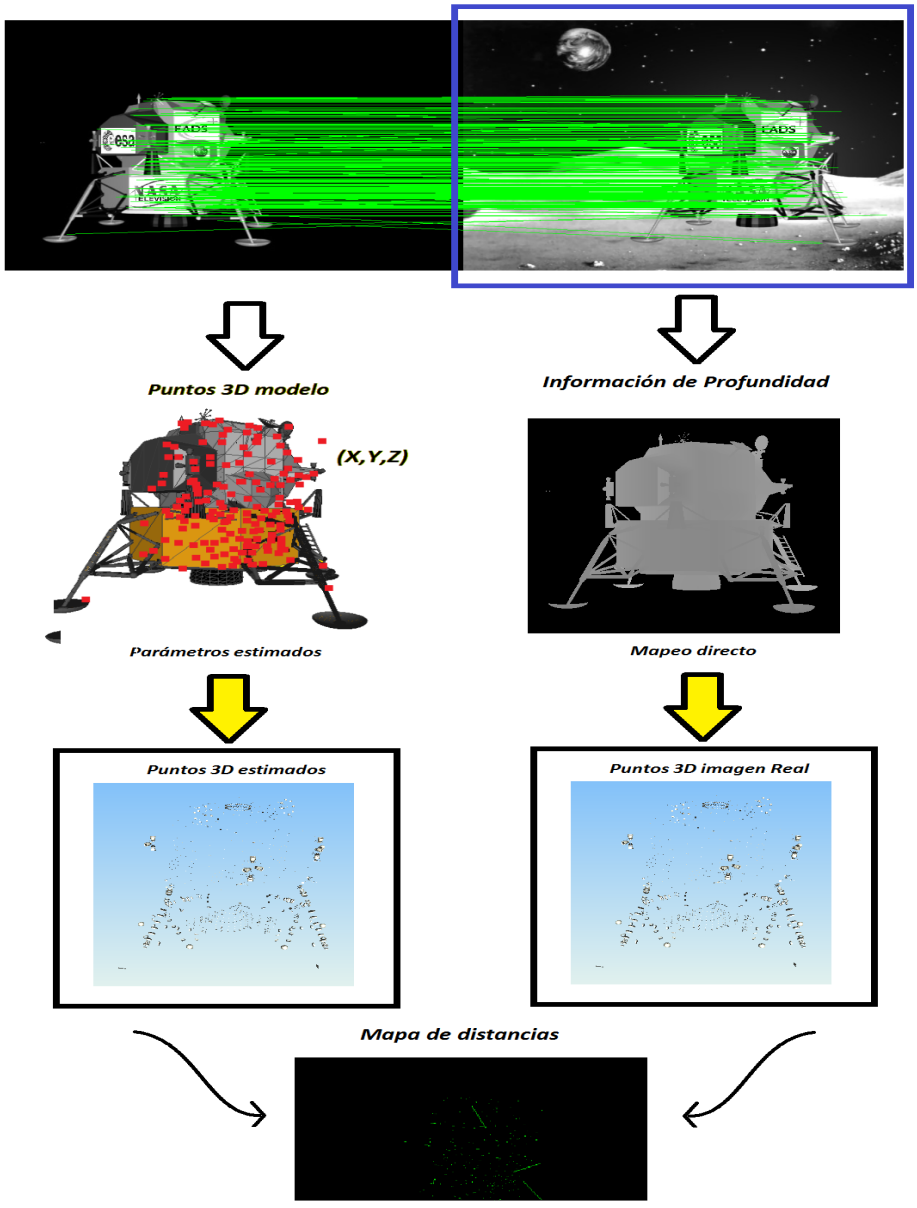
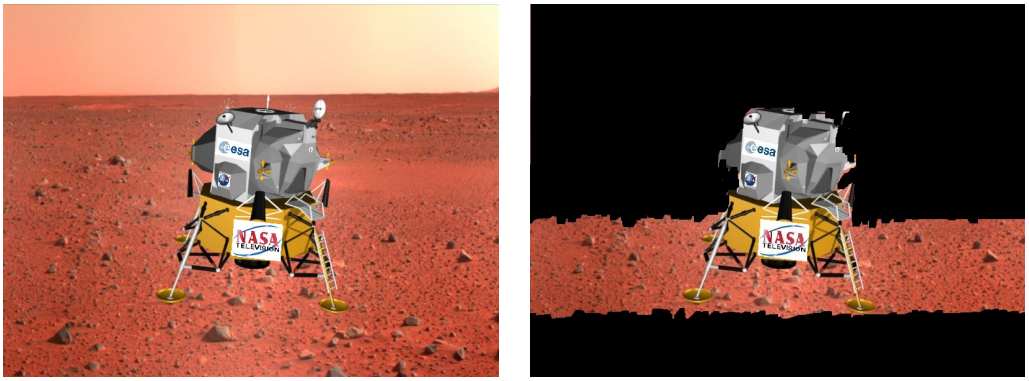


Figura 6.21: Cálculo de error parcial con información de profundidad. Como se puede comprobar en la figura, la cámara ToF ofrece un mapa de profundidad asociado a la imagen real. El procedimiento es similar al de la figura 6.19, si bien ahora se construyen mapas de puntos característicos 3D. Por último, se realiza el mapa de distancias en el espacio 3D y se calcula el error parcial como suma de todas ellas.



(a) (b)

Figura 6.22: Filtrado interno ToF. En (a) se presenta la imagen sin filtrar. En (b) se realiza filtrado interno especificando el rango de distancias en el que se sitúa el objeto LEM.

obtener información y el resto no se considera. Esta nueva etapa de filtrado, *interna a la cámara*, representa una ayuda más al objetivo de reducir el número de correspondencias erróneas que se producen y por lo tanto a mejorar las prestaciones del algoritmo global.

Levenberg Marquardt El método de optimización utilizado es Levenberg-Marquardt. La razón que determina su uso es el hecho de que sólo necesite la función error, un vector de observaciones o datos deseados y una estimación inicial para converger a la solución. Además, los resultados que ofrece para el problema de ajuste de mínimos cuadrados son robustos, convergiendo rápidamente en la mayoría de los casos. En el anexo C se realiza una descripción detallada del algoritmo.

6.5.3. Tratamiento de datos corruptos

En esta sección se plantean las alternativas de tratamiento de datos corruptos que definen los algoritmos mencionados en la introducción. La primera de ellas implementa una etapa de eliminación 3σ y la segunda RANSAC. Es importante destacar que ***ambas alternativas se apoyan en el algoritmo basado en mínimos cuadrados*** explicado con

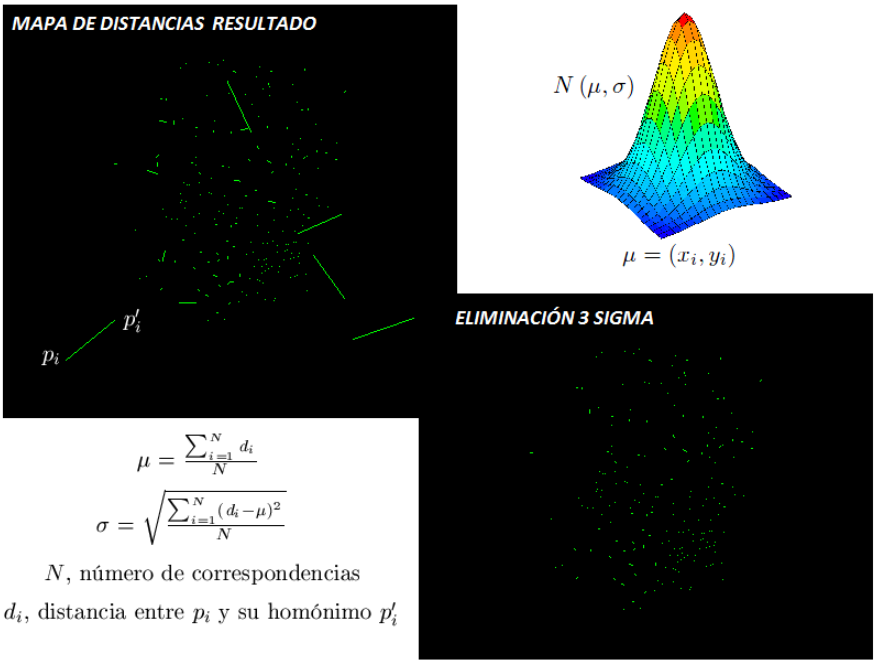


Figura 6.23: Etapa de eliminación 3σ.

anterioridad. En ningún caso afecta a POSIT. A continuación se analiza en detalle cada una de las alternativas.

6.5.3.1. Eliminación de outliers 3σ

Una vez realizada la primera fase de optimización, se procede con una etapa de eliminación de asociaciones incorrectas. El objetivo es la convergencia a un resultado más estable y próximo al deseado.

En la figura 6.23 se presenta el procedimiento completo. Se toma el *mapa de distancias resultado* de la etapa de optimización anterior y se calcula la media muestral μ, así como la desviación típica σ del conjunto de distancias. Se realiza un control sobre la variable de longitud, eliminando las distancias que residan fuera del rango $[\mu - 3\sigma, \mu + 3\sigma]$. El resultado es un nuevo mapa de correspondencias libre de datos corruptos, que conduce en general a una convergencia óptima del algoritmo.

En la figura 6.24 se presentan los resultados finales derivados de la

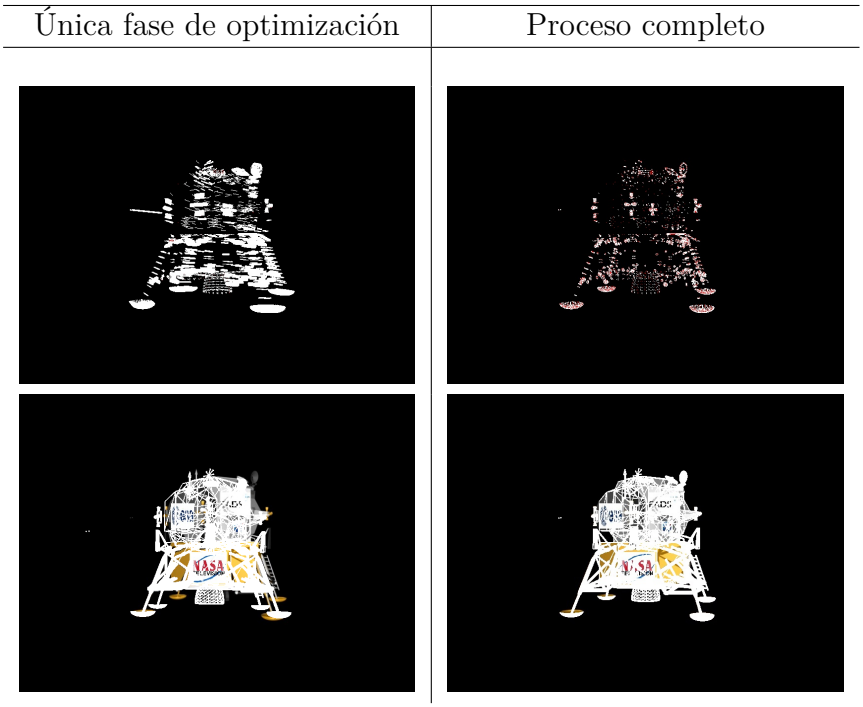


Figura 6.24: Resultados de error con/sin etapa de eliminación 3σ. En la columna de la izquierda se presentan los resultados de error tras la primera etapa de optimización. La columna de la derecha ilustra el error final del algoritmo. El error final se muestra en sus dos variantes como se explicará en la sección 6.5.4. Visualmente se comprueba como el error es menor tras la utilización de la etapa de eliminación 3σ (fijarse en las patas).

utilización o no de esta fase. Visualmente comprobamos como la incorporación de esta etapa mejora sustancialmente el error final obtenido.

Cabe remarcar que no siempre se produce una mejora derivada de la implementación de la fase de eliminación 3σ, ya que depende en gran medida de los parámetros que se definen en la etapa de filtrado y de correspondencia. Si las asociaciones son correctas, en general no mejora. En el resto de casos es fundamental para refinar la pose 3D del objeto.

6.5.3.2. RANSAC

Se implementa RANSAC tal como viene definido en la sección 4.7. Se introducen las siguientes mejoras:

- Número de iteraciones variable en función al contexto. De esta manera, se consiguen probabilidades de acierto similares, entorno al 99 % independientemente del estado en el que se encuentre el algoritmo. Además, reducimos el tiempo y coste computacional.

- Parámetro adicional de control, ζ. Su función es la de impedir que se produzcan soluciones determinadas por conjuntos de puntos reducidos. El valor de dicha variable se definió en torno al 30 % de datos de la muestra. En este sentido, se discriminan las soluciones parciales que ofrece el algoritmo y se producen mejoras en cuanto a precisión, eficiencia, tiempo y coste computacional.

El ajuste de los parámetros intrínsecos del algoritmo se reserva para la sección 6.6.1.

6.5.4. Cálculo del error final

En esta sección se introducen las dos formas de error final que se consideran:

- *Error tipo A: Mapa de distancias del conjunto de puntos del modelo 3D proyectados con pose estimada y real.* En la figura 6.25 se presentan varios ejemplos de mapas de error final. Los puntos en color rojo identifican estimaciones correctas. Un segmento blanco representa el error que se comete en la estimación. En la figura 6.25 se presentan soluciones con error ascendente.

- *Error tipo B: Proyección del modelo sobre la imagen real, error visual.* Es la otra alternativa de visualización del error. En la imagen 6.25 se muestra un ejemplo de aplicación sobre escenario lunar y fondo homogéneo. En el ejemplo, el error se acentúa de manera constante hasta alcanzar un máximo sobre la última figura.

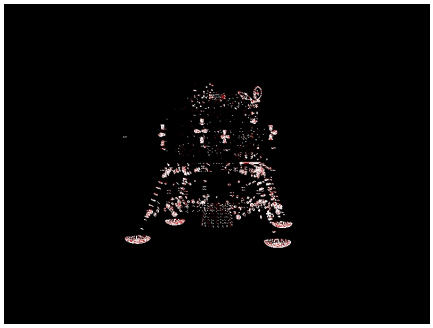
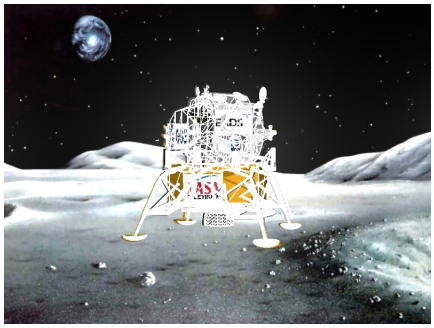
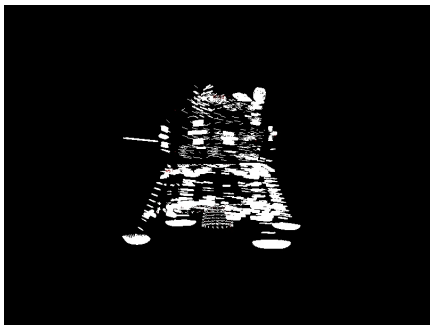
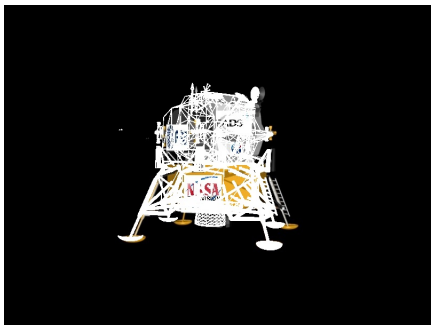
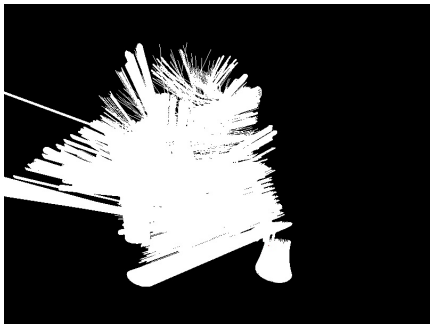
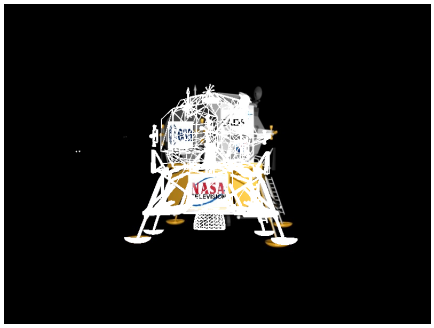
Error Tipo A	Error Tipo B
	
	
	

Figura 6.25: Error total.

El error final se calcula a partir de todos los puntos del modelo 3D como sigue:

$$E_{TOTAL} = \frac{1}{N}\sum_{i=1}^{N} \triangle p_i$$

donde N es el número total de puntos del modelo $3D$ y $\triangle p_i$ es la longitud del segmento que une el punto i-ésimo del modelo proyectado con la pose estimada y su correspondiente con la pose real. El error total representa *la distancia media en número de pixels que separa la pose real de la estimada.*

6.6. Seguimiento

En esta sección se realiza una introducción al *tracking* 3D del objeto. Aunque no se considera como objetivo inicial del proyecto, se realiza una pequeña aproximación para así poder entender y valorar las propiedades y características del algoritmo implementado.

En secciones anteriores se desarrollaban las distintas etapas del algoritmo en función al objetivo marcado, refinamiento de la pose 3D de un objeto. Sin embargo, si deseamos realizar tracking o seguimiento debemos estar preparados para escenarios en los que el objeto se *"pierde"*, reside fuera del entorno visual o aparece pero no somos capaces de identificarlo, caso de estar lo suficientemente alejado. En este sentido, tenemos en cuenta las siguientes consideraciones:

- *Caso Objeto Perdido.* Existen tres posibles motivos para llegar a esta situación.

    - El primer caso sucede cuando un objeto externo interfiere en la imagen, ocultando parte o la totalidad de la estructura del objeto bajo estudio. Como consecuencia, se produce una pérdida de información, directamente relacionada con el número de correspondencias y el algoritmo no converge.

- El segundo caso se produce cuando el objeto se desplaza a una pose lo suficientemente alejada, fuera de los límites de la etapa de filtrado. No se establecen correspondencias, o las que se generan son incorrectas y el algoritmo deja de converger.
- El último caso sucede cuando el algoritmo no converge aún teniendo un número suficiente de correspondencias.

En todos los casos anteriores se mantiene la estimación de parámetros de pose resultado de la última convergencia del algoritmo. Si el método no converge en un número I de iteraciones, entonces el objeto se da por perdido y se produce la desactivación de los filtros diseñados en la primera parte del algoritmo. De esta manera, se buscan correspondencias en toda la imagen con la esperanza de encontrar asociaciones correctas.

- *Caso Objeto encontrado.* Si el algoritmo converge para la imagen actual, se almacenan los parámetros asociados a la pose 3D estimada y *se utilizan como inicialización de pose para la imagen siguiente.* De esta manera, se reduce el tiempo de convergencia y se orienta al algoritmo en la dirección correcta de búsqueda.

En la figura 6.26 se presenta gráficamente el funcionamiento del algoritmo de tracking implementado.

6.6.1. Parámetros RANSAC

En la sección 4.7.1 se realizaba un estudio sobre los distintos parámetros que intervienen en RANSAC. En este caso, se distinguen los siguientes escenarios:

- *Objeto Encontrado.* Entendemos como valor extremo el 25 % de outliers presentes en la muestra. Haciendo uso de la tabla 4.1, el número mínimo de iteraciones es aproximadamente 50 (considerando 8 correspondencias en media).

- *Objeto Perdido*. Tomamos como valor extremo entorno al 40 % de outliers en la muestra. Haciendo uso de la tabla 4.1, el número mínimo de iteraciones es 250 (considerando 8 correspondencias en media).

Por lo tanto, el número de iteraciones es variable y dependiente del contexto.

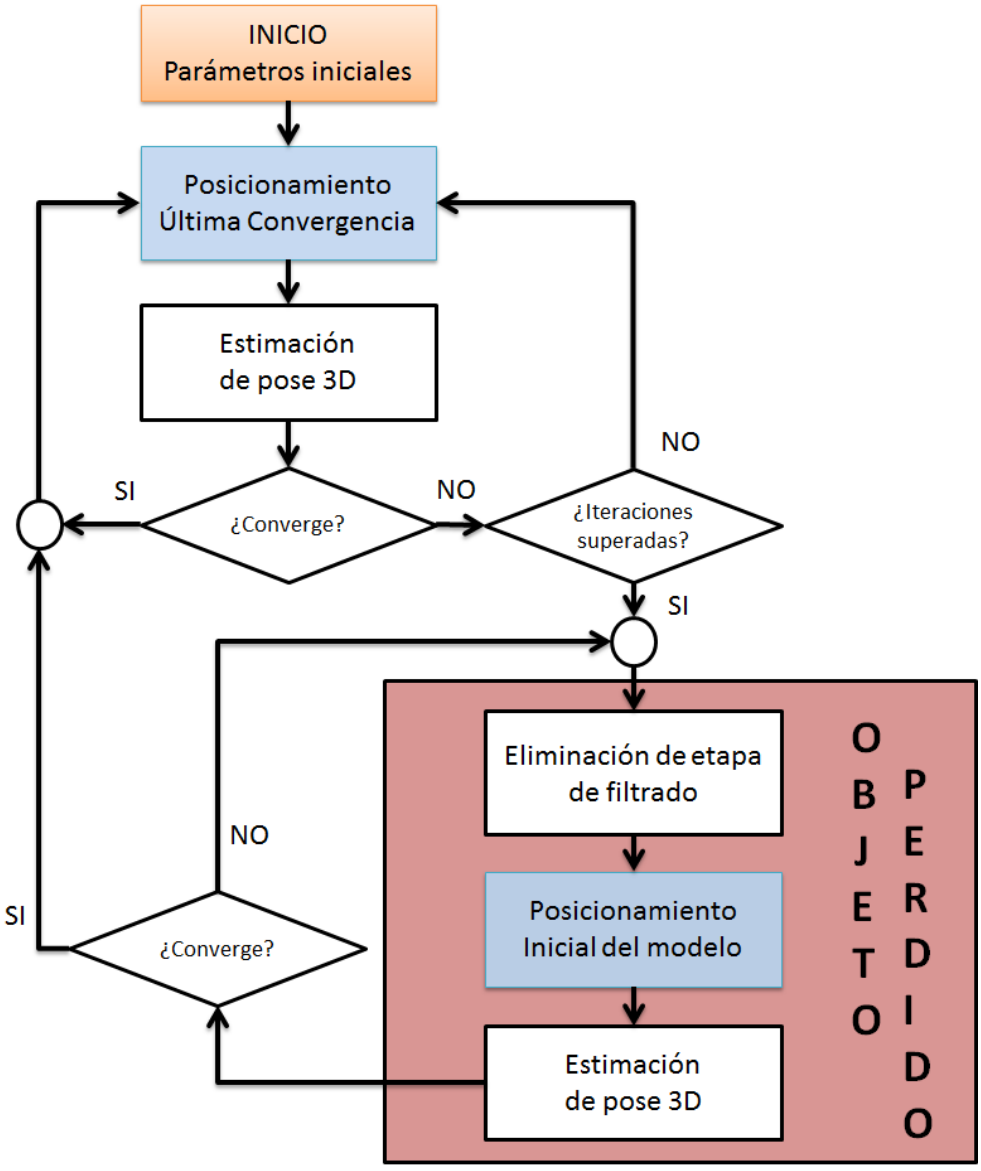


Figura 6.26: Descripción del algoritmo de tracking.

Capítulo 7

Experimentos

7.1. Introducción

En este capítulo se analizan las prestaciones y características más relevantes del conjunto de algoritmos implementado. Además, se presentan las secuencias de tracking realizadas y las conclusiones más significativas.

El capítulo se estructura en dos grandes bloques. En el primer bloque se presentan los modelos y datos generados, así como la terminología necesaria. En el segundo bloque se presentan los resultados generales de aplicación.

7.2. Modelos y datos utilizados

En esta sección se detalla el conjunto de modelos y datos utilizados en el desarrollo del proyecto.

7.2.1. Datos necesarios

Los modelos $3D$ utilizados pertenecen a las siguientes familias:

- *Modelos reales*. Objetos reales de los cuales se obtiene el modelo 3D asociado.

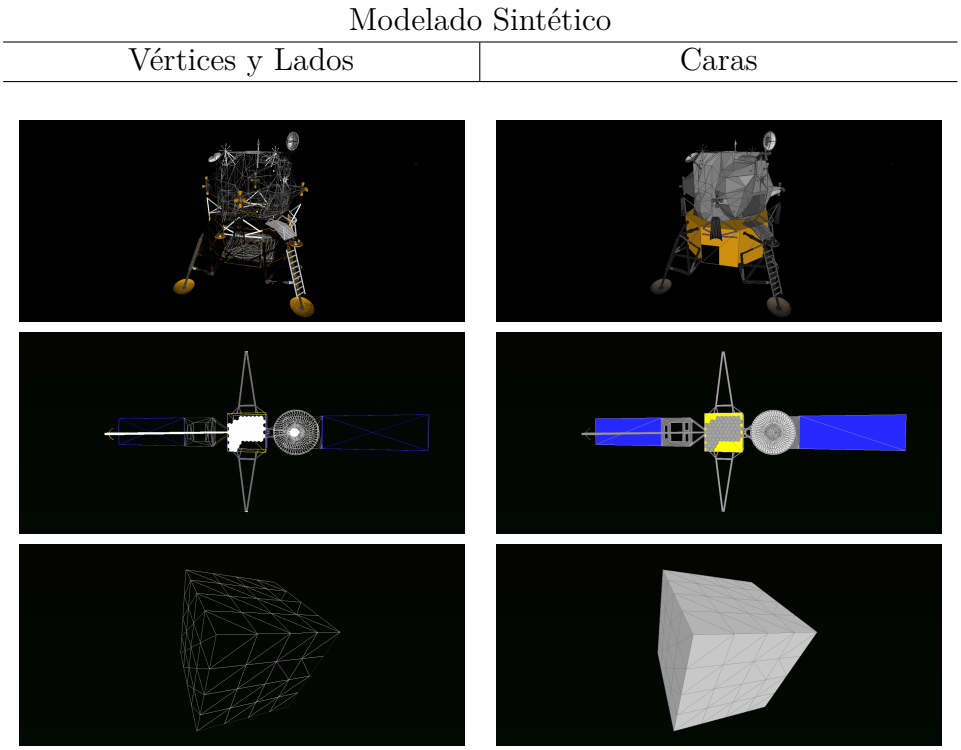


Figura 7.1: Modelos sintéticos 3D desarrollados (LEM, satélite y cubo).

- *Modelos sintéticos, artificiales*. Objetos diseñados por ordenador. En este sentido se dispone de toda la información referente al objeto, así como sus parámetros de pose en la imagen.

A continuación se presentan las características más relevantes de ambas alternativas.

7.2.1.1. Modelado artificial

El modelado artificial permite simular todo tipo de condiciones reales de visibilidad sobre el objeto. Entre los parámetros más importantes destacan los referentes a la distorsión en la cámara, condiciones de visibilidad, luminosidad, etc. Además, permite, en el caso de no disponer de un sistema calibrado de visión, la obtención de resultados numéricos cualitativos para valorar la eficacia y eficiencia del algoritmo implemen-

Modelado Real

Vértices Caras

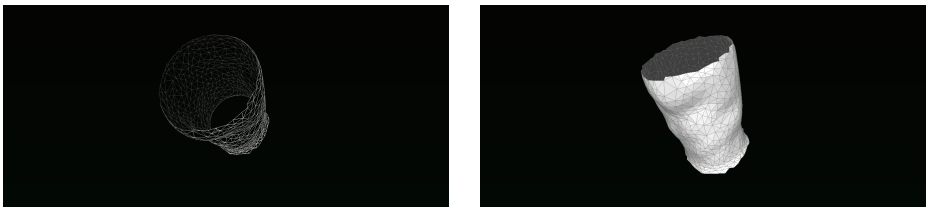


Figura 7.2: Modelo real BACI.

tado.

A continuación, se presenta el conjunto de objetos sintéticos 3D desarrollados:

- Módulo lunar LEM. Modelo utilizado como guía en la explicación de los algoritmos desarrollados en el capítulo 6.
- Satélite de comunicaciones.
- Cubo.

En la figura 7.1 se presenta la estructura de caras y vértices correspondiente a cada modelo.

7.2.1.2. Modelado real

El objeto real seleccionado es un tarro, BACI, cuyo modelo 3D se presenta en la figura 7.2. El método utilizado de reconstrucción utiliza un conjunto de imagenes del objeto tomadas desde diferentes perspectivas [31]. Como se puede comprobar, la fase de modelación 3D no ofrece resultados completamente correctos, y el modelo 3D resultante tiene pequeñas deformaciones en la estructura de sus caras.

7.2.2. Imagenes modelo utilizadas

En el capítulo 4 se introdujo el conjunto de datos previos necesarios a la ejecución del algoritmo. En general, se exponían dos conjuntos, el

Imagenes referencia modelo 2D

Figura 7.3: Imagenes referencia 2D de modelos sintéticos y real.

formado por los modelos 3D de los objetos y el de las imágenes 2D de referencia. En esta sección se introducen las imágenes 2D asociadas a los modelos 3D ya explicados.

La imagen modelo debe presentar el objeto sobre fondo uniforme. De esta manera, estamos seguros que en la fase de correspondencias, el conjunto de puntos resultantes pertenecen al objeto y por lo tanto al modelo 3D asociado. En la figura 7.3 se presentan las imágenes modelo base para los cuatro objetos utilizados.

Por último, recordar que las imagenes de referencia $2D$ van acompañadas de los parámetros de pose $3D$ del objeto.

Escenarios: Satélite de comunicaciones

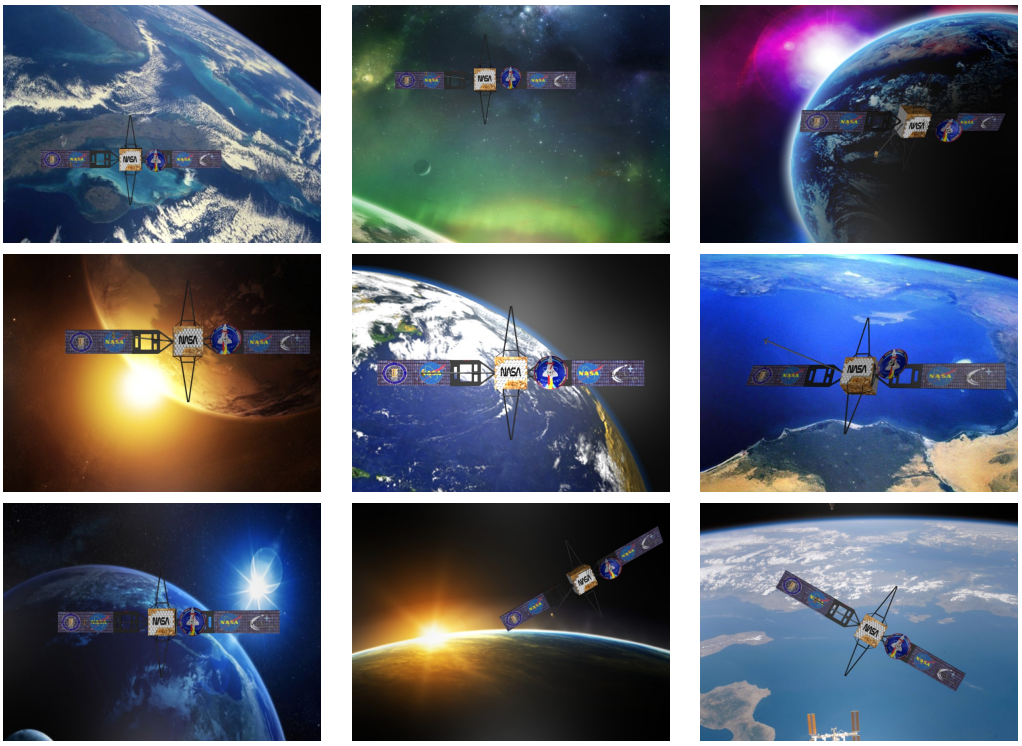


Figura 7.4: Escenarios satélite.

7.2.2.1. Escenarios

Con el objetivo de realizar una caracterización completa de los algoritmos implementados, hemos realizado el estudio sobre diferente tipo de escenarios.

El escenario determina de manera directa la capacidad de convergencia del algoritmo, representando una forma de ruido añadida que reduce, según su tipología, en mayor o menor medida el número de asociaciones correctas que se producen.

En la figura 7.4 se presentan los escenarios utilizados en el proceso de reconocimiento del satélite de comunicaciones. En la figura 6.12 (capítulo 6, *algoritmos desarrollados*) se muestran los escenarios utilizados para detectar el módulo lunar LEM.

Escenario: BACI

Figura 7.5: Escenario BACI.

En la figura 7.5 se presenta el escenario real en el que se encuentra el objeto BACI (laboratorio). Como se puede apreciar, el contexto es una mesa sobre la que se disponen diversos objetos con el objetivo de valorar la robustez del algoritmo. Así mismo, se mueve el objeto de manera manual obteniendo los resultados que se presentan en la sección de seguimiento.

7.2.2.2. Mapas de profundidad

En la sección 6.5.2.1 se propuso la inclusión de la profundidad como información añadida. A continuación se presentan dos escenarios posibles en función al tipo de modelo utilizado:

- *Modelado virtual.* Se generan los mapas de profundidad de forma

Mapas de profundidad

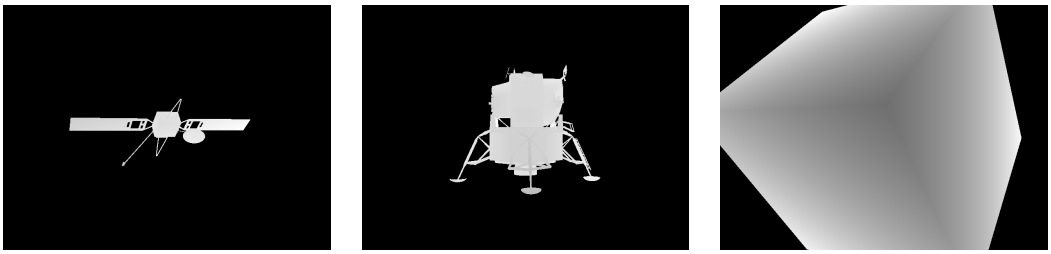


Figura 7.6: Mapas de profundidad: modelos sintéticos.

matemática vía Blender. Esto es posible ya que se conoce la pose del objeto en la imagen real. La proyección del modelo con los parámetros de pose conocidos permiten generar el mapa de profundidad de manera directa. En la figura 7.6 se presentan ejemplos de mapas de profundidad relativos a los modelos virtuales del satélite, módulo lunar LEM y cubo.

- *Mapas de profundidad en tiempo real con cámara ToF.* En la figura 7.7 se presenta el conjunto de información que da como resultado la cámara ToF utilizada.

7.2.2.3. Objetos interferentes 3D

Como colofón a la fase de caracterización del conjunto de datos utilizados, y con el objetivo de realizar un estudio más general, se introducen nuevos modelos $3D$ que ejecutan el rol de objeto interferente. De esta manera, conseguimos contextos de visualización parcial y podemos definir el error en estas circunstancias. En la figura 7.8 se ilustran diversos ejemplos de visión parcial.

7.2.3. Representación del error total

En el capítulo 6 se realizaba un análisis completo sobre el error final resultado del algoritmo. A modo de recordatorio, se definen dos tipos de error:

Imagenes Cámara ToF

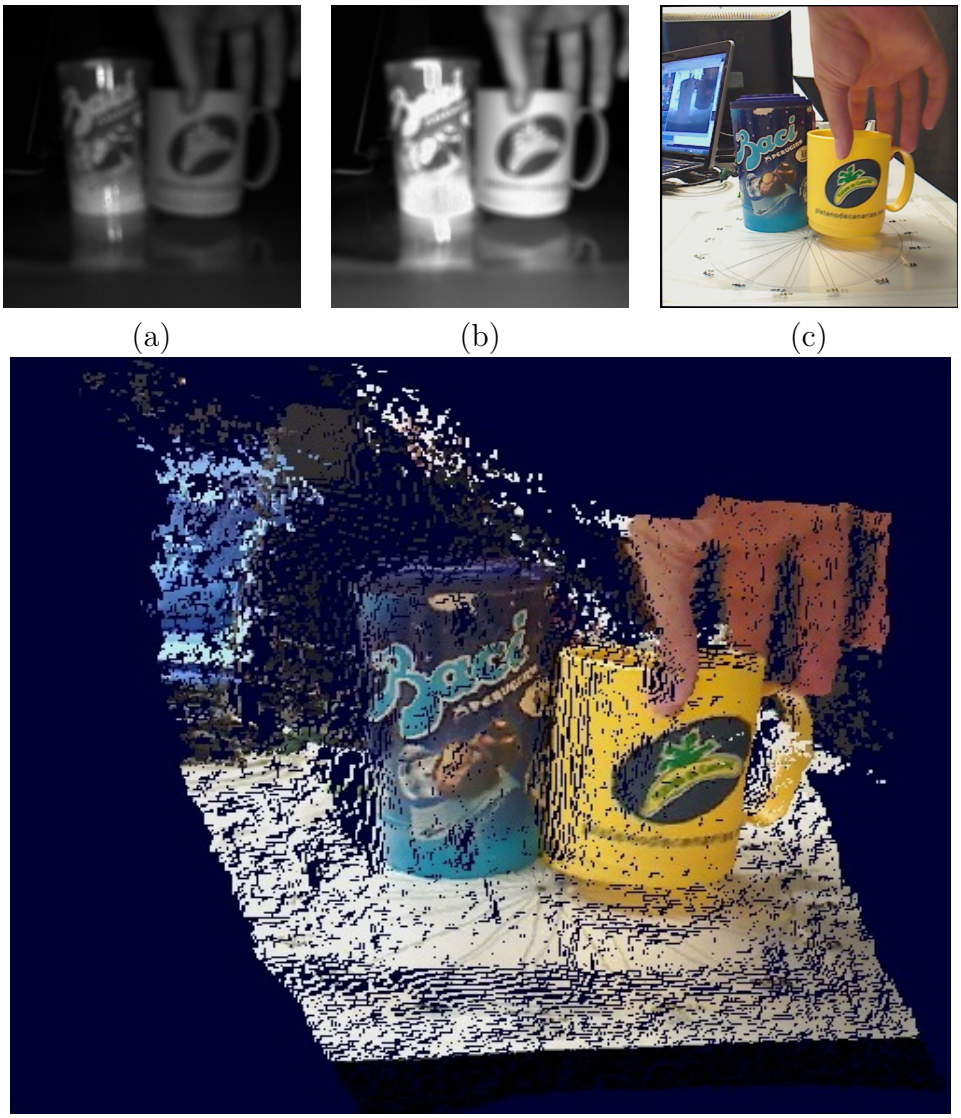


Figura 7.7: Imagen completa BACI con cámara ToF. En las figuras a), b) y c) se presentan las imagenes de amplitud, intensidad y color. En la figura d) se ilustra el mapa de profundidad asociado a la toma.

Visión Parcial

Figura 7.8: Objetos interferentes 3D. Escenarios de visión parcial.

- Error numérico (tipo A). Representado mediante un mapa de distancias del conjunto de puntos imagen real y estimados. Permite catalogar numéricamente el resultado. Sin embargo, *tiene el inconveniente de que se precisa disponer de un sistema calibrado de visión para poder realizar los cálculos correspondientes*. Es decir, necesitamos conocer la pose del objeto en la imagen real para así poder compararla con la estimada. Por este motivo, ***realizamos el estudio de error numérico sobre el conjunto de objetos artificiales en exclusiva***. En la figura 7.9 se presenta, a modo de ejemplo, diversos resultados de este tipo de error sobre a los modelos satélite, LEM y cubo.

- Error visual (tipo B). Proyección directa del modelo 3D, con los parámetros de pose estimados sobre la imagen real. Representa una

Error Tipo A sobre Objetos Sintéticos

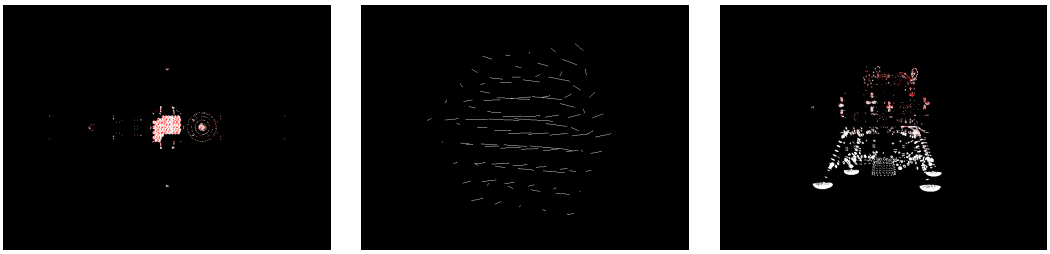


Figura 7.9: Error tipo A final.

segunda alternativa de representación del error final. En este caso, no se precisa del conocimiento de los parámetros de pose finales, por lo tanto es una ***técnica aplicable a los dos conjuntos de datos: modelos reales y sintéticos***. En la aplicación de tracking se utilizan de manera conjunta para obtener información en tiempo real del error cometido. En la figura 7.10 se presentan varios resultados de este tipo de error sobre el conjunto de datos utilizados.

7.3. Resultados

En esta sección se analizan las prestaciones y características más relevantes del conjunto de algoritmos implementado. Para ello, se realiza la caracterización del error ante transformaciones espaciales (rotación sobre ejes y translación) y contextos de ruido variable. Además, se evalúa la eficacia en función del porcentaje de datos corruptos y se presentan los resultados asociados a la fase de seguimiento.

7.3.1. Transformaciones espaciales

Las transformaciones estudiadas son:

- Rotación sobre eje X.
- Rotación sobre eje Y.

Error tipo B sobre imagenes sintéticas

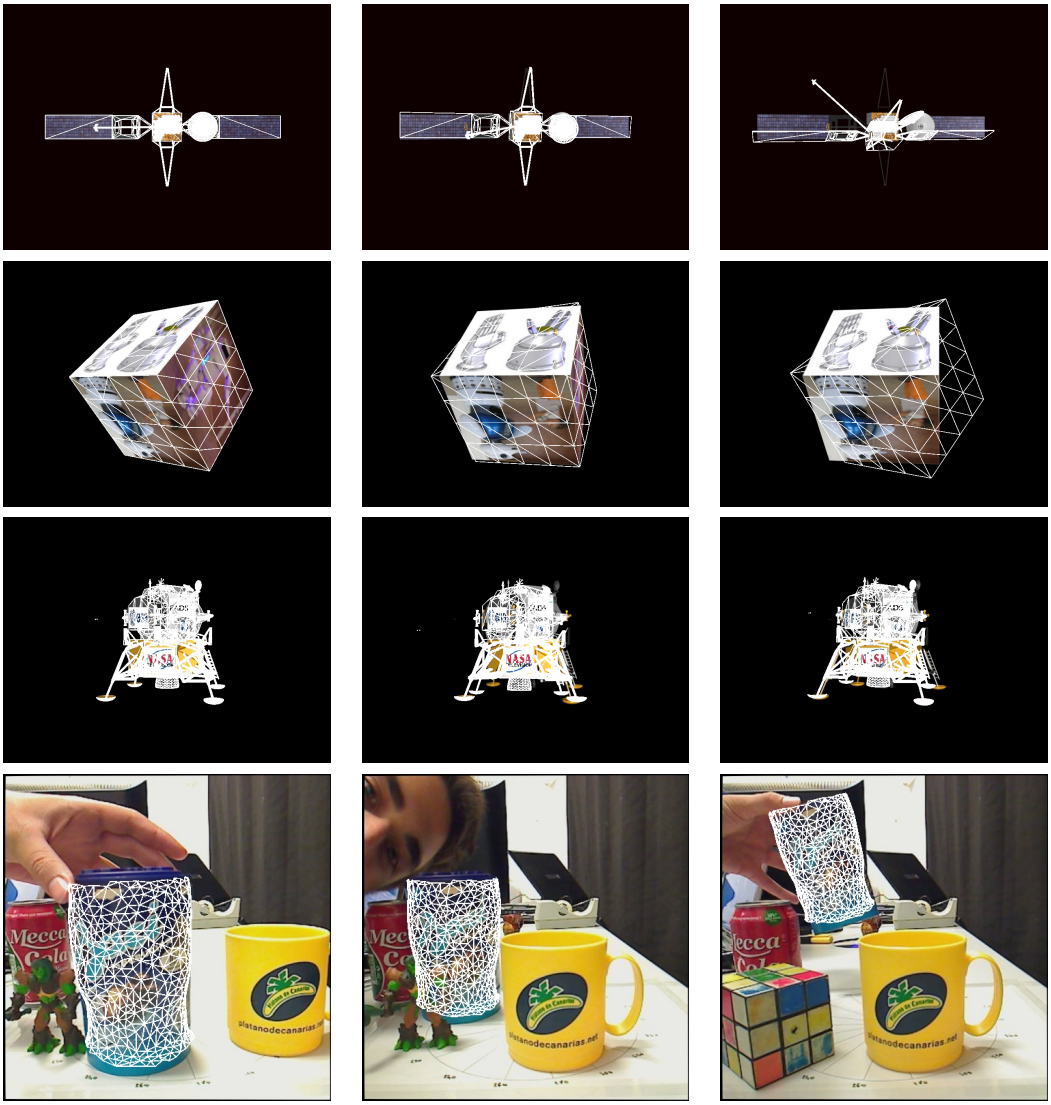


Figura 7.10: Error final tipo B.

Imagen Real	Correspondencias	Error Visual

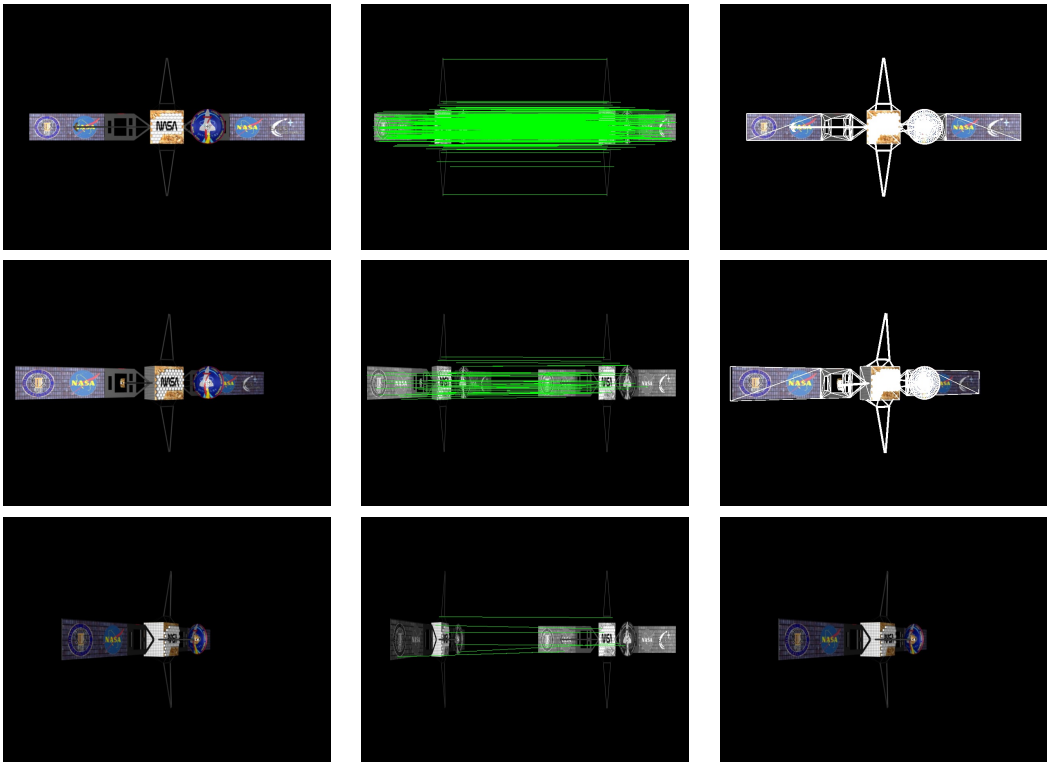


Figura 7.11: Error de rotación sobre eje Y, RMS.

- Rotación sobre eje Z.
- Translación.

7.3.1.1. Rotación sobre eje Y

En esta sección se analiza el comportamiento del error ante una rotación de α grados en torno al eje Y. El objetivo es determinar el *rango máximo de variación* en el que se obtienen resultados aceptables para el conjunto de algoritmos implementados.

En la figura 7.11 se presenta el estudio de error asociado al modelo satélite. A medida que el ángulo de rotación α crece, el número de asociaciones disminuye y el porcentaje de falsas correspondencias se incrementa.

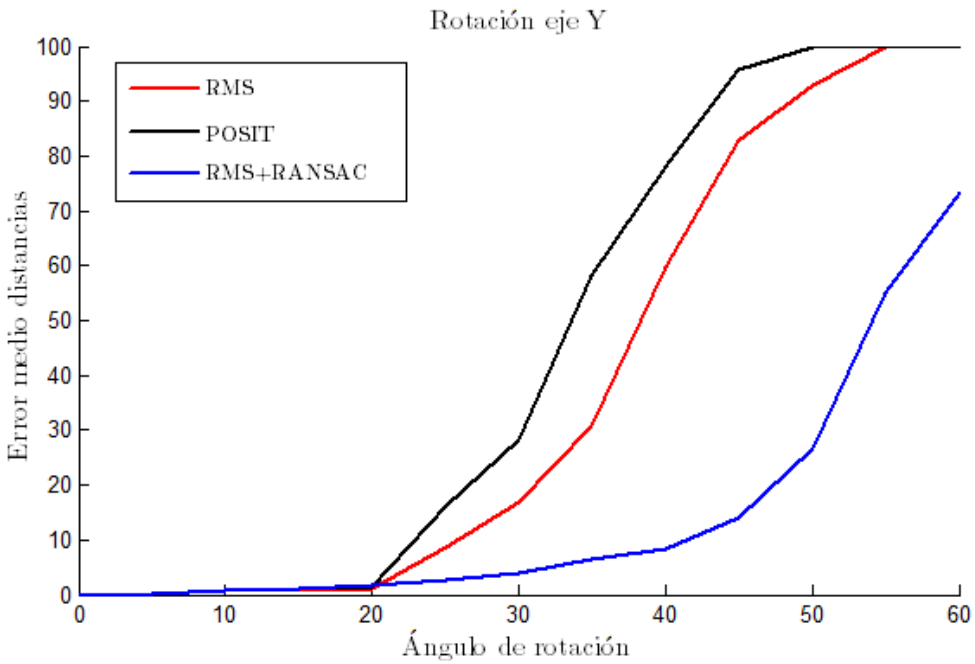


Figura 7.12: Rotación sobre eje Y.

La consecuencia directa es un aumento del error final.

La figura 7.12 presenta la variación del error medio en función del ángulo de rotación sobre el eje Y para el conjunto de algoritmos implementados. Todos los métodos presentan resultados de error similares hasta ángulos de rotación de 20^{o}. A partir de ese instante se producen variaciones significativas entre ellos:

- POSIT es el algoritmo que peores resultados ofrece. En otras palabras, es el algoritmo que peor interactúa en presencia de datos erróneos. A continuación se posiciona el algoritmo RMS que mejora levemente los resultados ofrecidos por POSIT.

- La incorporación de RANSAC al algoritmo RMS produce una mejora sustancial de la capacidad de convergencia del algoritmo. El tiempo de cómputo se incrementa, sin embargo, es capaz de obtener resultados válidos incluso en escenarios alejados. En este sentido

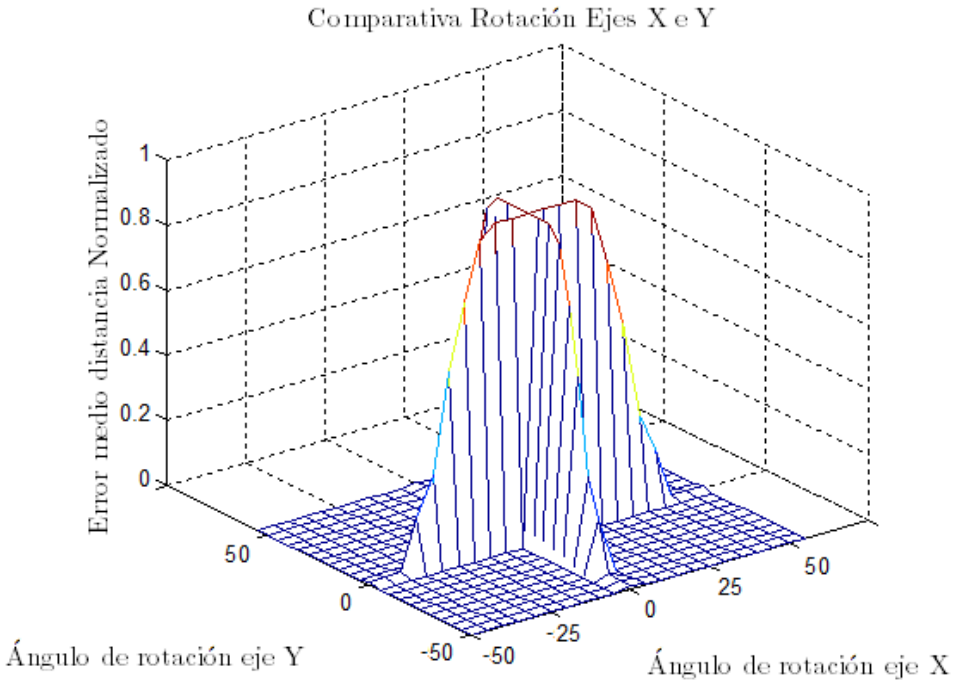


Figura 7.13: Comparativa de rotación sobre ejes X e Y.

tenemos una relación de correspondencia entre *tiempo de cómputo y rango de funcionamiento*.

7.3.1.2. Rotación sobre eje X

Los resultados de rotación sobre eje X son equivalentes a los resultados presentados en la sección anterior. Este resultado concuerda con la teoría de geometría espacial ya que la parte de la imagen que varía (puntos del modelo visibles) en ambas transformaciones es equivalente. En la figura 7.13 se ilustra la relación de equivalencia entre ambas transformaciones.

7.3.1.3. Rotación sobre eje Z

En esta sección se presentan los resultados de rotación según el eje Z. A diferencia de lo acontecido en las transformaciones de rotación an-

Imagen Real	Correspondencias	Error Visual

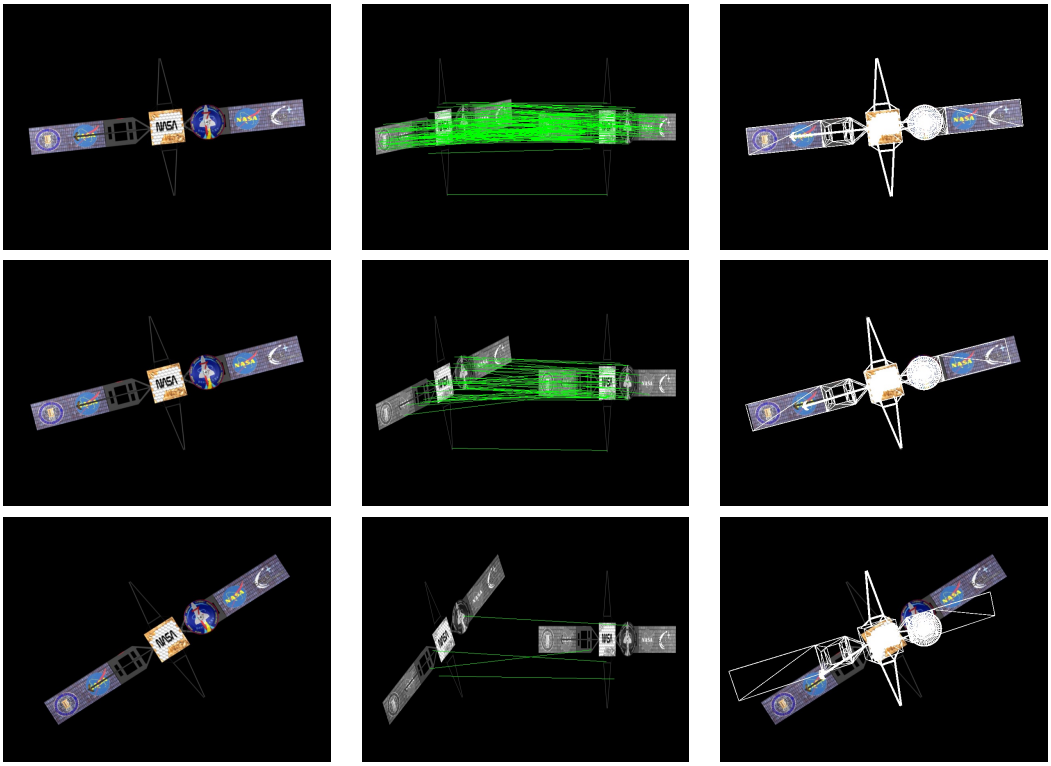


Figura 7.14: Rotación sobre el eje Z, RMS.

teriores, la parte del objeto que visualizamos permanece constante. Lo único que se modifica es la orientación de los puntos característicos en la imagen.

En la figura 7.14 se presenta el efecto de dicha rotación sobre el modelo satélite utilizando el método RMS para estimación de pose. La estructura visible del objeto permanece constante. Sin embargo, debido a la etapa de filtrado así como a la variación de parámetros característicos del punto de interés, el número de asociaciones disminuye a medida que crece el ángulo β de rotación sobre dicho eje.

En la figura 7.15 se presentan los resultados de error del conjunto de algoritmos implementados ante este tipo de transformación. Se obtienen las siguientes conclusiones:

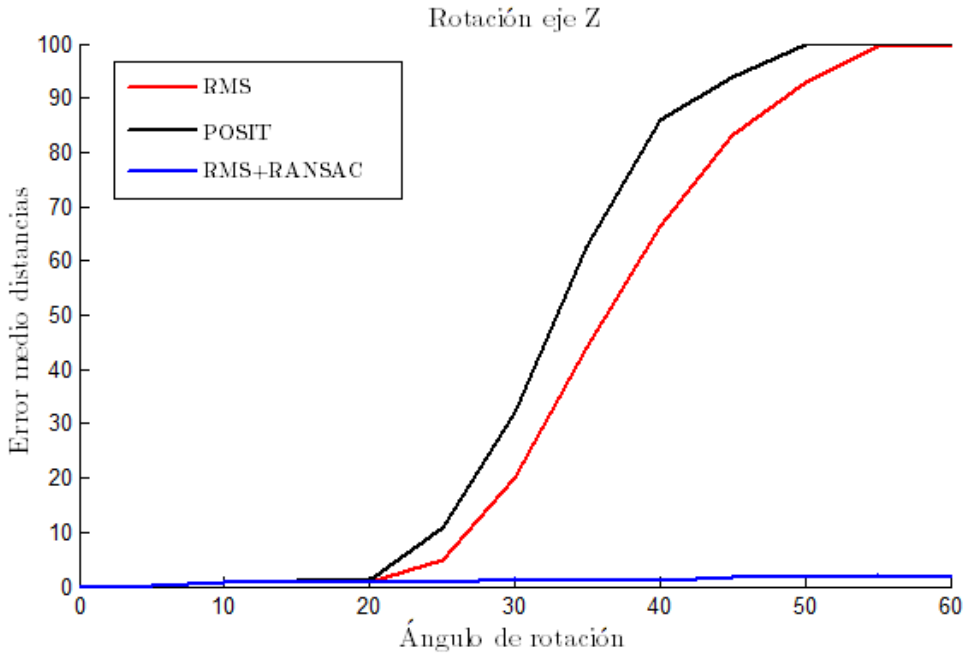


Figura 7.15: Rotación sobre eje Z.

- POSIT y RMS se comportan de manera similar a lo acontecido en las rotaciones sobre los ejes X e Y.

- La incorporación de RANSAC, con etapa de filtrado inicial desactivada, mejora de manera sustancial los resultados obtenidos por los métodos anteriores en solitario. Además, presenta resultados de error inferiores a los de rotación sobre el resto de ejes.

7.3.1.4. Translación

En esta sección se caracteriza la evolución del error ante transformaciones de translación. De manera similar a la transformación de rotación sobre el eje Z, si sólo se produce translación, la parte de la estructura del objeto visible permanece constante.

En la figura 7.16 se presentan los efectos de translación sobre eje Z utilizando como referencia el modelo satélite de comunicaciones. El método

Imagen Real	Correspondencias	Error Visual

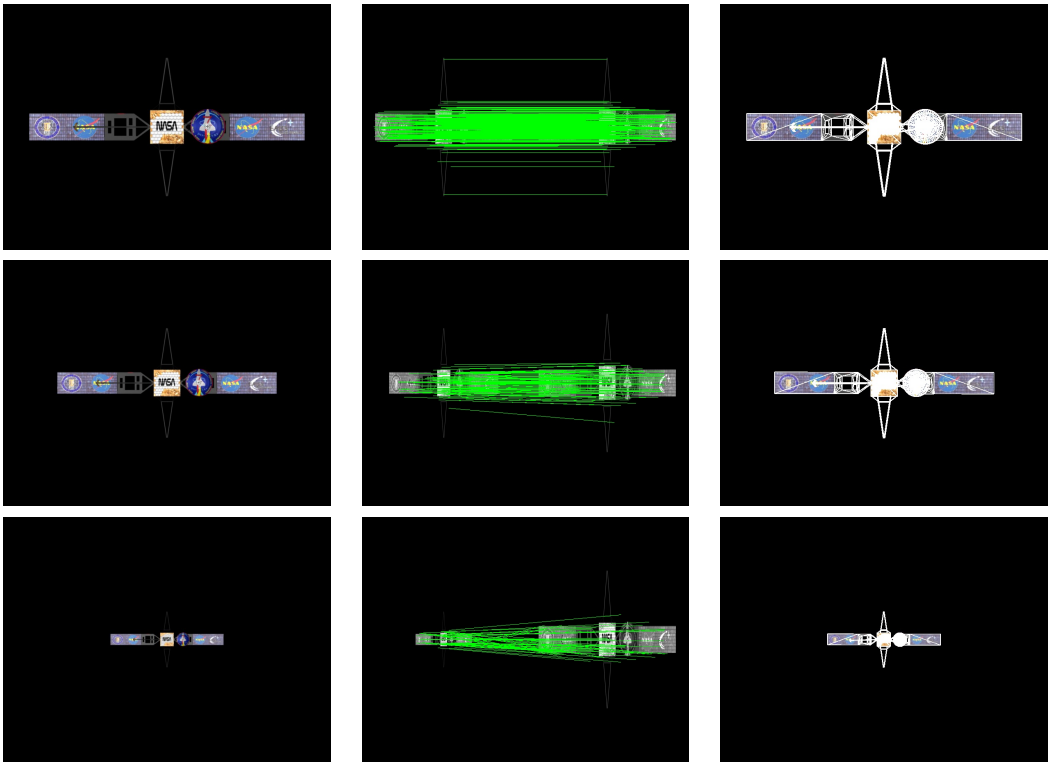


Figura 7.16: Error de translación, RANSAC+RMS

de estimación seleccionado es RANSAC en conjunción con RMS. Como se puede observar, el número de asociaciones desciende lévemente a medida que se incrementa la escala de translación. Sin embargo, no se produce un descenso considerable de dicho parámetro ya que la información que arrojan los descriptores de puntos en ambas imágenes son similares. En este contexto, la limitación viene determinada por la etapa de filtrado inicial. El factor limitante es el radio característico asociado al pixel de interés, ya que el resto de parámetros permanecerían constantes. En el caso de realizar transformaciones de translación sobre el eje X o Y, el factor limitante es el filtro espacial.

RANSAC es el método que mejor interactúa ante este tipo de transformación. Algoritmos en solitario como POSIT y RMS presentan problemas

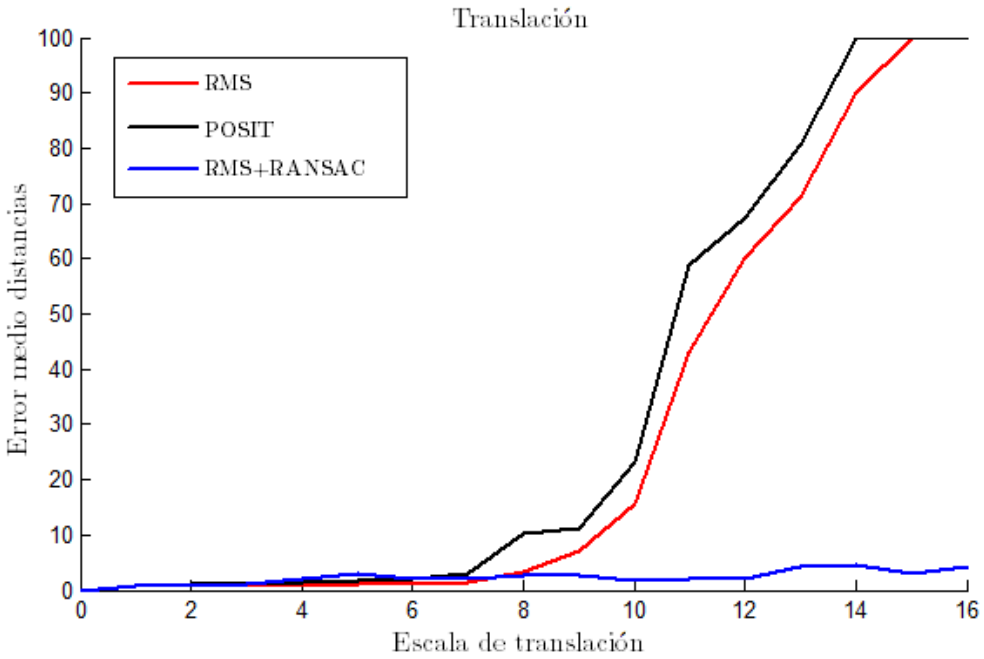


Figura 7.17: Translación media sobre eje Z. Escala 1:4. En este sentido el valor de escala intermedio 8 se corresponde con la visualización del objeto a una distancia tal que su tamaño sea $1/2$ del original.

de convergencia. Sin embargo, el rango es aceptable teniendo en cuenta el contexto de refinamiento de pose en el que nos encontramos.

En la figura 7.17 se ilustra la evolución del error ante una transformación de translación sobre el eje Z.

7.3.2. Escenarios Específicos

Es esta sección se estudian los siguientes escenarios:

- Ruido
- Eficacia en función del porcentaje de asociaciones correctas.

7.3.2.1. Ruido

En este apartado se caracterizan los efectos de la aparición de ruido sobre la imagen real. El ruido se modela mediante una distribución gaussiana de media nula y varianza σ^2. La transformación del pixel de coordenadas (x, y) viene determinada por la siguiente expresión:

$$I(x, y) = I(x, y) + \mathcal{N}\left(0, \sigma^2\right)$$

donde $I(x, y)$ es el valor del pixel de coordenadas (x, y) y $\mathcal{N}(0, \sigma^2)$ representa la variable aleatoria de ruido.

En la figura 7.18 se presenta el efecto de aparición de este tipo de ruido sobre la imagen satélite. A medida que aumenta la varianza del ruido introducido, se incrementa el número de puntos característicos. Esta circunstancia produce una disminución drástica del número de correspondencias resultante. Sin embargo, aún siendo bajo el nivel de asociaciones, la calidad de las correspondencias es buena, lo que produce resultados aceptables.

En la figura 7.19 se presentan los resultados de error para el conjunto de algoritmos estudiado en el caso particular de ruido guassiano. La tendencia permanece constante, es decir:

- El algoritmo que peor interactúa ante el ruido es POSIT, seguido de RMS.
- La incorporación de RANSAC mejora los resultados previos. Sin embargo, la diferencia entre ellos no es tan acusada como en otro tipo de transformaciones.

En definitiva los resultados obtenidos son aceptables, con índices de error inferiores a 2 pixels hasta una varianza de 60, lo que indica que este tipo de implementación es ***robusta al ruido***.

7.3.2.2. Porcentaje de inliers

En esta sección se presenta la respuesta de los algoritmos implementados ante la presencia de outliers en el conjunto de correspondencias.

SURF Imagen Real	Correspondencias	Error Visual

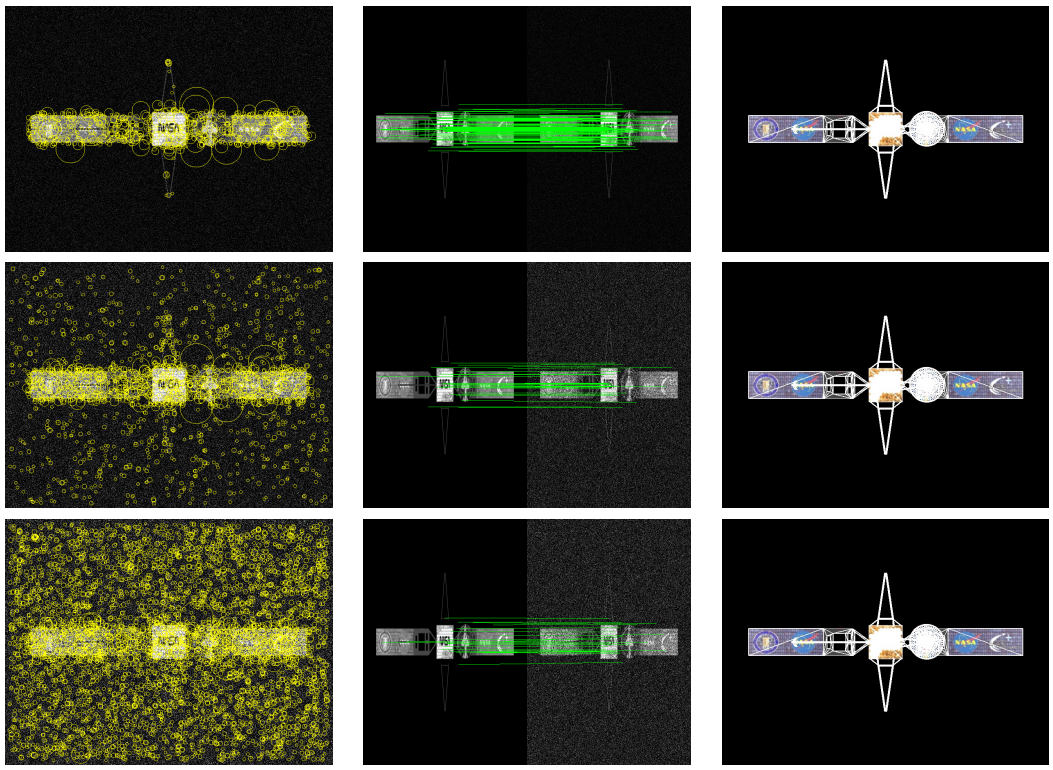


Figura 7.18: Ruido.

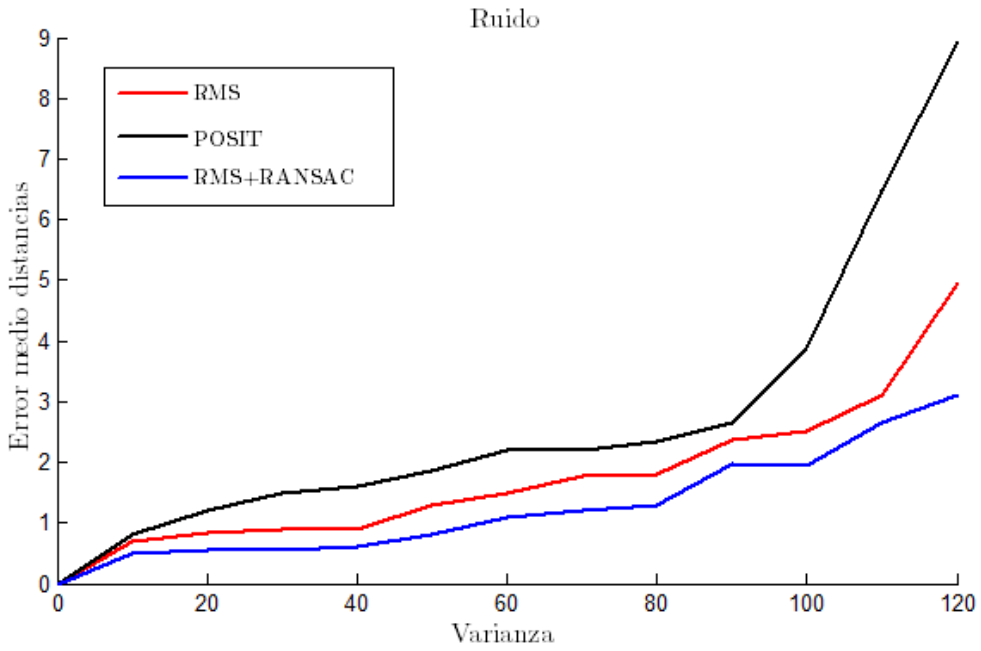


Figura 7.19: Ruido.

Los resultados obtenidos son acordes con las pruebas ya realizadas, ver figura 7.20. La curva converge con error mínimo para muestras con al menos el $50 - 60\,\%$ de datos correctos. Si el porcentaje es inferior, el error aumenta rápidamente y se producen problemas de convergencia. RANSAC en unión con RMS se consolida como el algoritmo más robusto.

7.3.3. Tracking

En esta sección se presentan las secuencias de tracking utilizadas y se analizan los resultados del algoritmo de seguimiento implementado.

7.3.3.1. Secuencias

Los modelos utilizados para las secuencias de tracking son el satélite de comunicaciones (secuencia sintética) y BACI (secuencia real). Con el objetivo de realizar un control total sobre el desarrollo del proceso, se

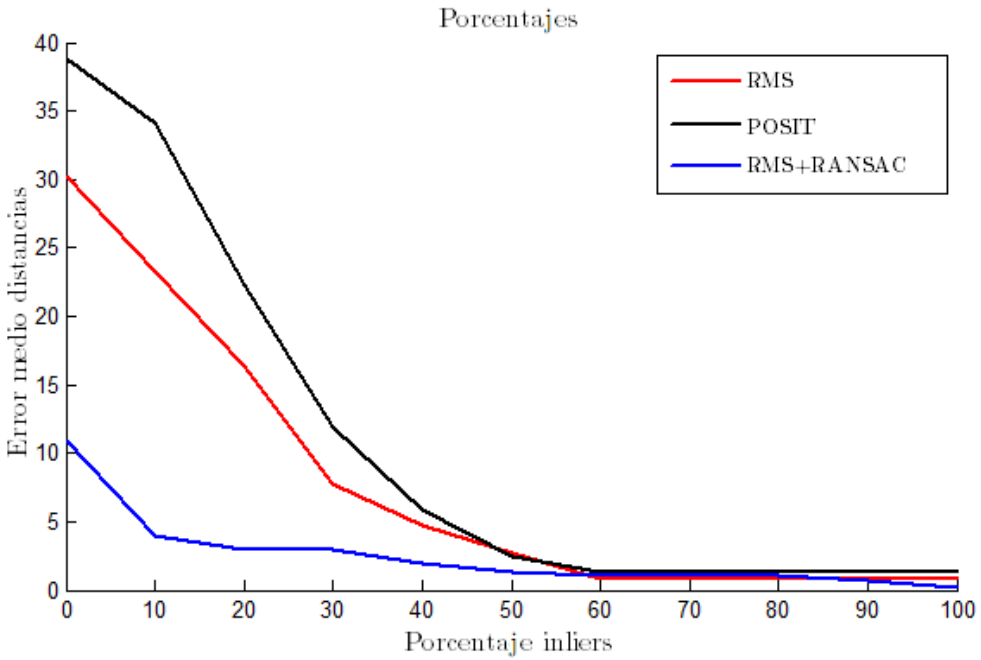


Figura 7.20: Porcentaje inliers.

implementa una ***plataforma visual*** que permite controlar en tiempo real la aportación de cada una de las etapas al resultado final. Además, ofrece la posibilidad al usuario de modificar, de manera interactiva, los distintos parámetros del algoritmo y observar sus consecuencias.

La plataforma visual muestra la siguiente información:

- Etapa de extracción de puntos característicos (figuras 7.24, 7.25).
- Etapa de correspondencias (figuras 7.26, 7.27).
- Error visual *tipo B* (figuras 7.28, 7.29).
- Error *tipo A*.
- Información adicional tal como el mapa de profundidad (figuras 7.22, 7.23).

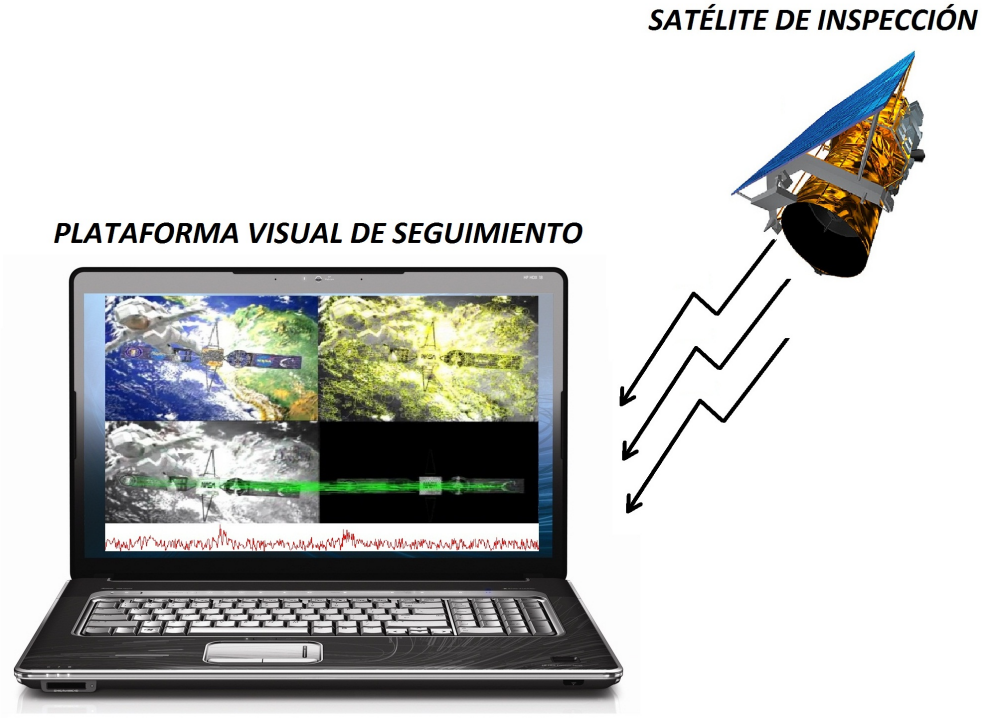


Figura 7.21: Plataforma visual de seguimiento. En la imagen se presenta la transmisión de información entre el satélite de inspección y la plataforma de seguimiento. En la parte superior de la pantalla se muestran las capturas de error visual y etapa de puntos característicos. En la parte inferior se presenta la etapa de correspondencias y error numérico relativo a la estimación de pose.

La información anterior se muestra en tiempo real por pantalla tal como se ilustra en la figura 7.3.3.1, si bien en esta sección se presentan las distintas componentes por separado.

Secuencia Tracking 3D BACI: profundidad

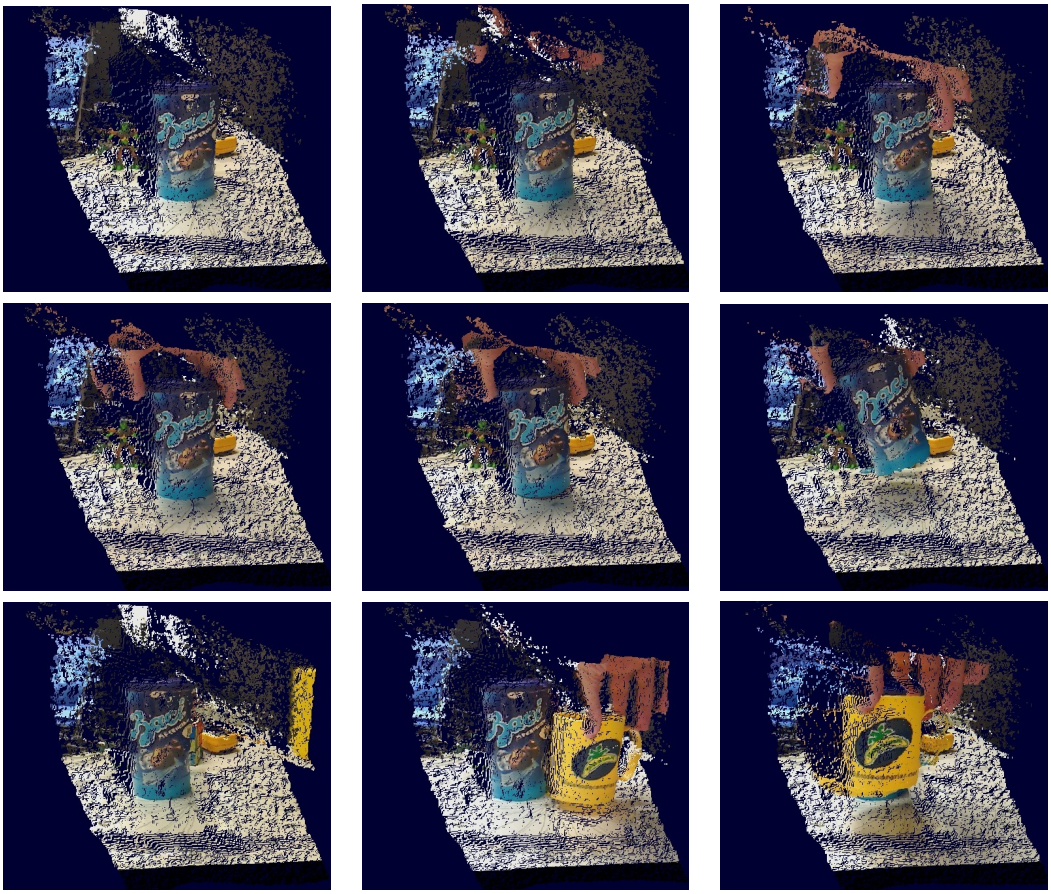


Figura 7.22: Información de profundidad sobre secuencia de tracking BACI.

Secuencia Tracking 3D Satélite: profundidad

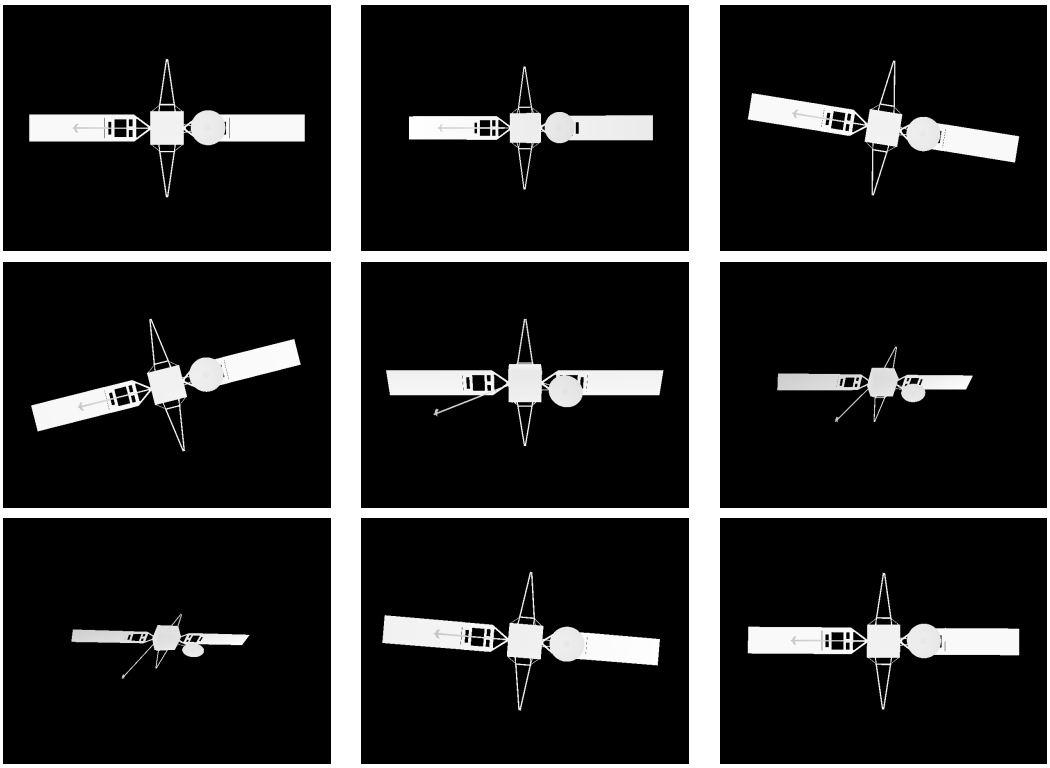


Figura 7.23: Mapas de profundidad sobre secuencia de tracking satélite.

Secuencia Tracking Satélite: puntos de interés

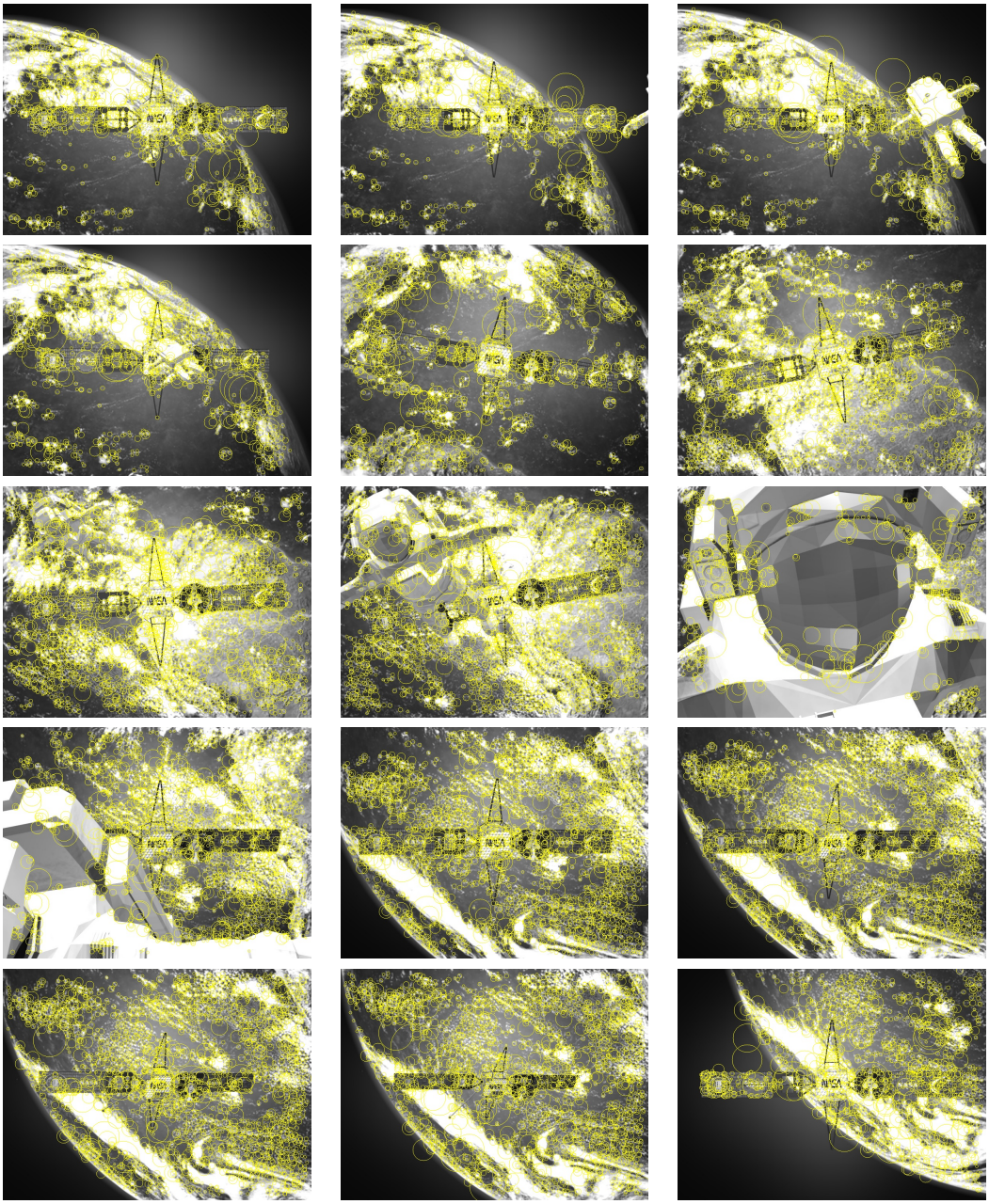


Figura 7.24: Puntos de interés sobre secuencia de tracking satélite.

Secuencia Tracking BACI: puntos de interés

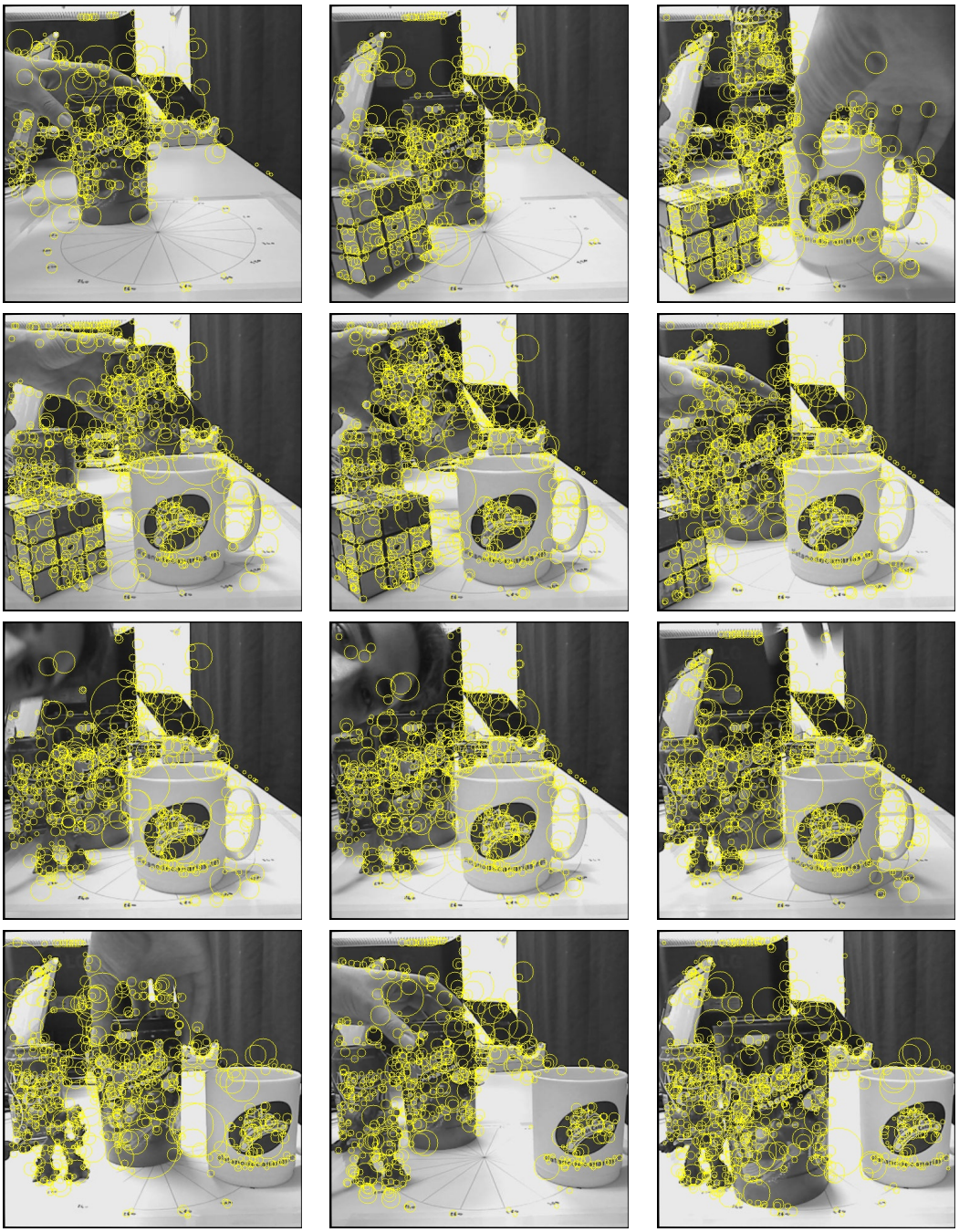


Figura 7.25: Puntos de interés sobre secuencia de tracking BACI.

Secuencia Tracking Satélite: correspondencias

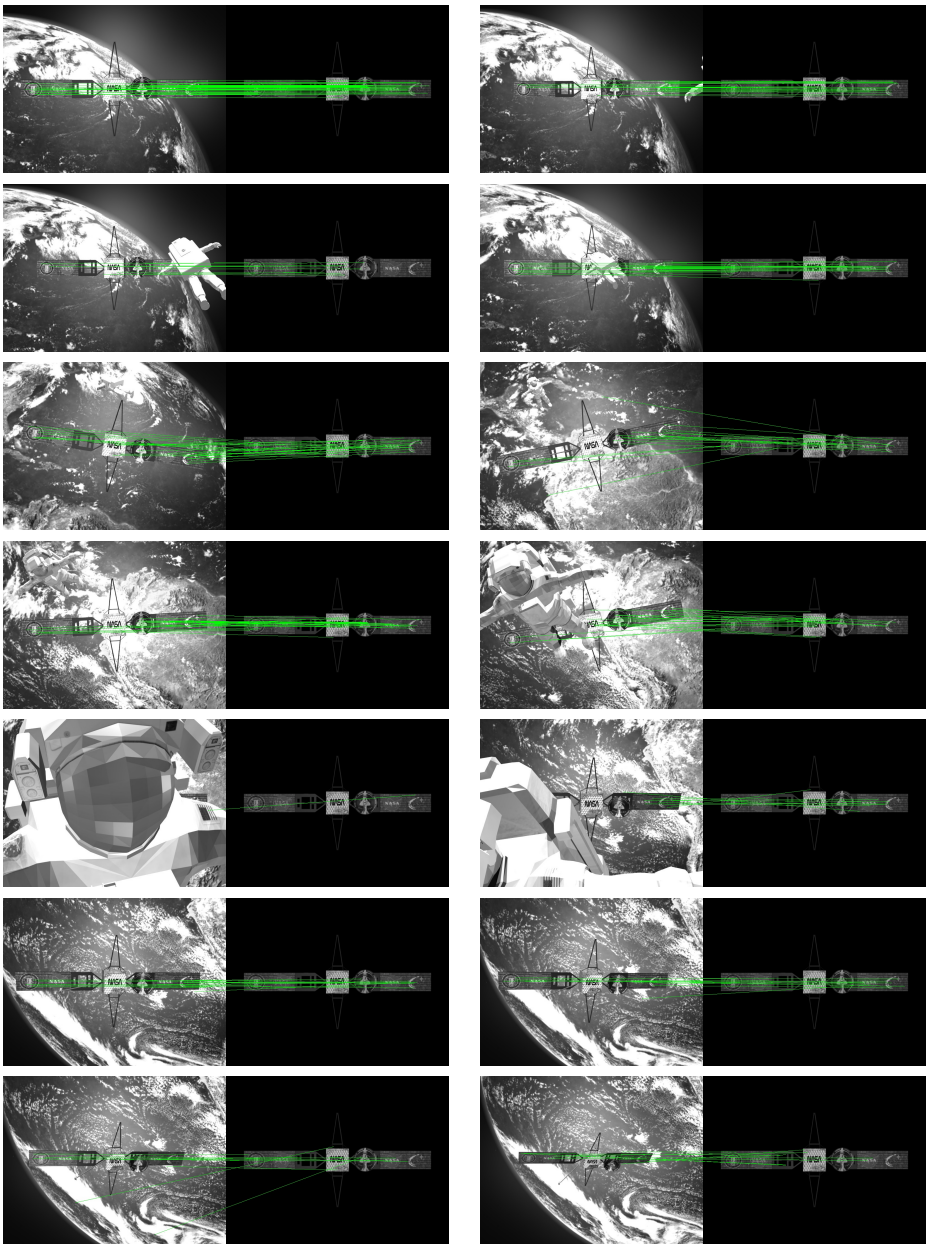


Figura 7.26: Correspondencias sobre secuencia de tracking satélite.

Secuencia Tracking BACI: correspondencias

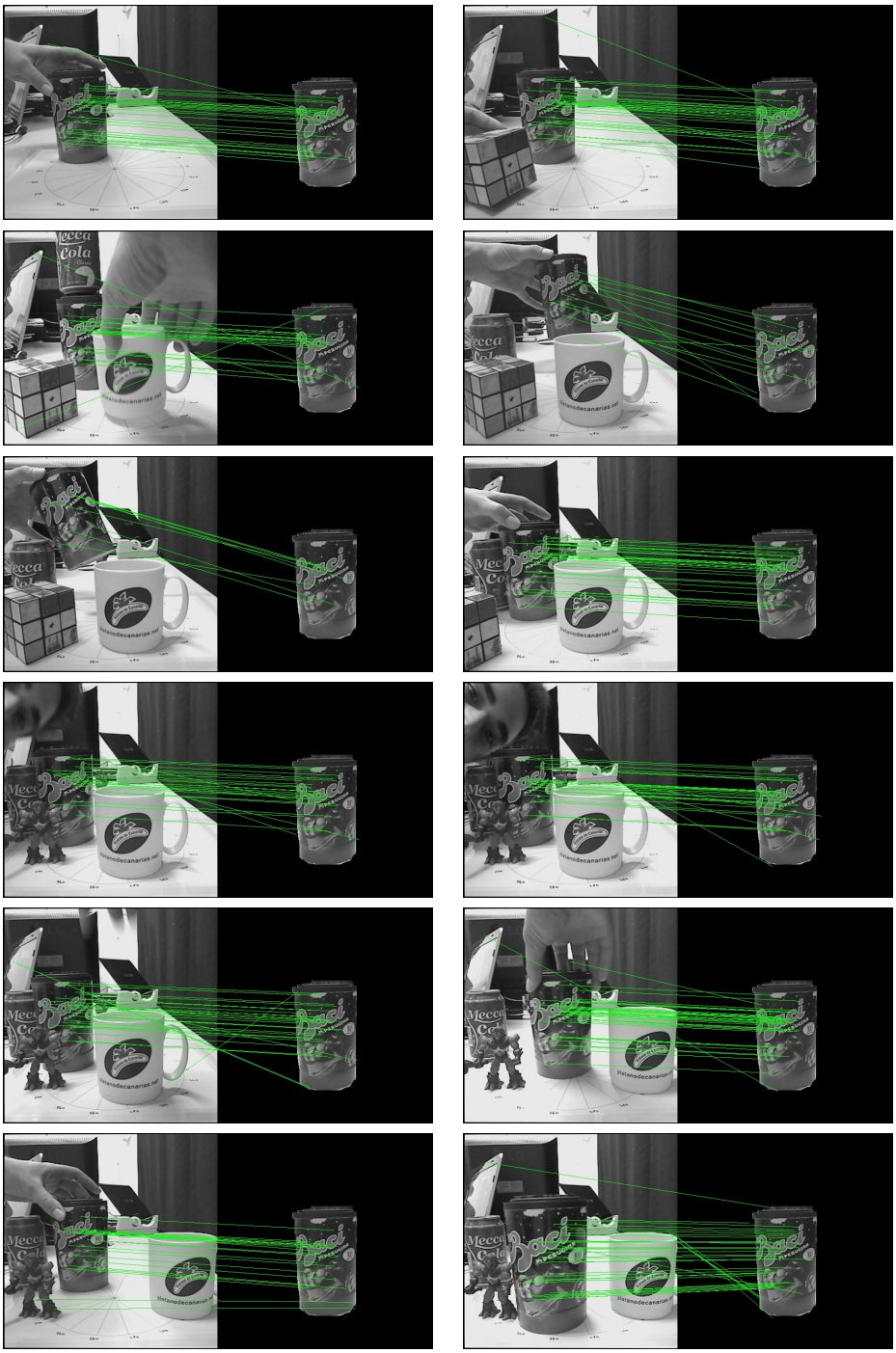


Figura 7.27: Correspondencias sobre secuencia de tracking BACI.

Secuencia Tracking Satélite: error final

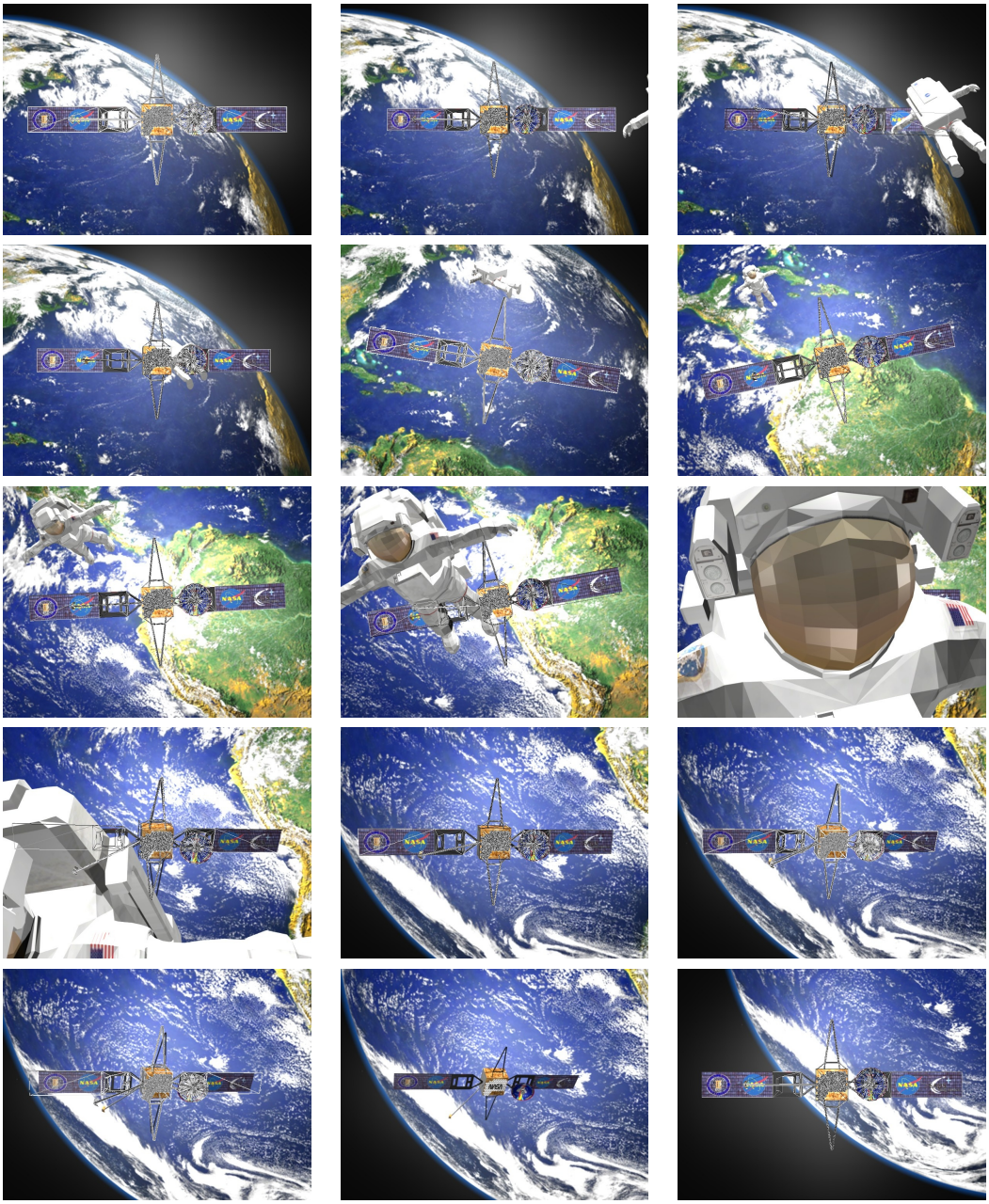


Figura 7.28: Error final sobre secuencia de tracking satélite.

Secuencia Tracking BACI: error final

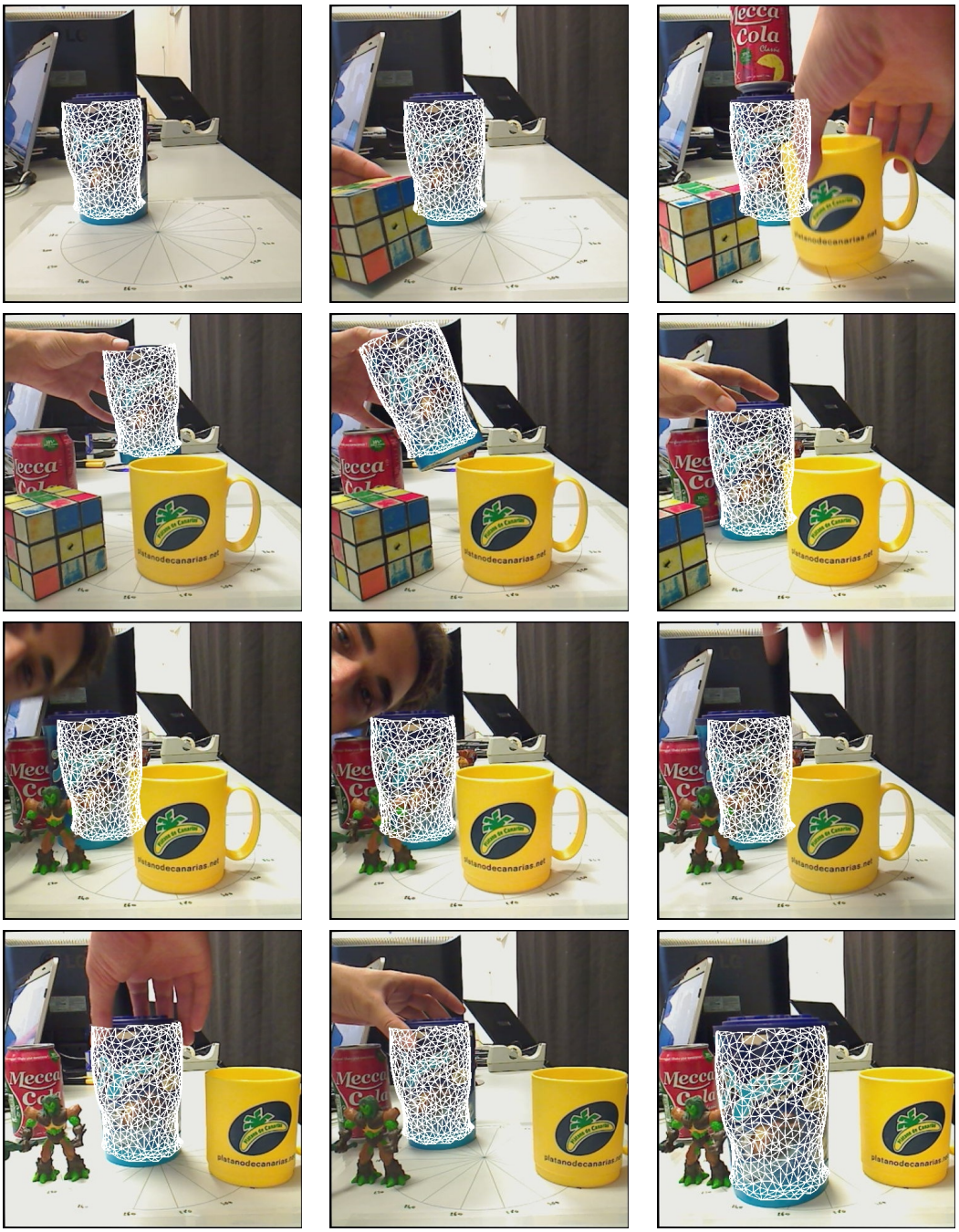


Figura 7.29: Error final sobre secuencia de tracking BACI.

7.3.3.2. Resultados Tracking

En la figura 7.30 se presentan los resultados de error relativos a la secuencia de tracking del satélite de comunicaciones para el conjunto de algoritmos implementado. Las conclusiones principales son:

- *Presencia de outliers.* POSIT es el algoritmo que más se ve a afectado por la presencia de datos corruptos. En la gráfica se observa a partir de la varianza que se produce en la estimación de pose. Observamos como para frames consecutivos la estimación varía significativamente. Por lo tanto, pequeñas variaciones en el conjunto de datos utilizados, varían la convergencia del algoritmo de forma significativa.
  Como era de esperar, el algoritmo basado en mínimos cuadrados reduce la varianza introducida por POSIT. La aplicación de RANSAC a RMS termina por optimizar dicho parámetro, obteniéndose resultados de error mínimos.

- *Error medio.* En concordancia con los resultados presentados en la sección 7.3.1 y 7.3.2, el algoritmo que mayor error presenta es POSIT, seguido de RMS y RANSAC.

- *Objeto alejado.* En los instantes A y B de la figura 7.30 se ilustra la situación en la que el satélite varía su pose de manera significativa respecto la pose inicial. Los algoritmos POSIT y RMS sufren una variación en el error medio pero mantienen su varianza. Dicha variación es superior en POSIT y visualmente se comprueba observando la interferencia de señales de error en ambos sectores. Como era de esperar, el algoritmo RANSAC permanece inalterable.

- *Objeto perdido por interferencia.* Situación ilustrada en el sector C de la figura 7.30. A medida que el astronauta interfiere en la imagen, la información visual del satélite disminuye. En este sentido, se produce una reducción paulatina del número de asociaciones y de la validez de las mismas. Llega un punto en el que la información visual es tan reducida que el algoritmo no converge y pasa al estado

objeto perdido, manteniendo los parámetros de pose de la última convergencia durante un tiempo determinado. Además se desactiva la etapa de filtrado inicial y el algoritmo intenta identificar correspondencias en toda la imagen. En la figura 7.31 se presenta dicha secuencia. Como era de esperar, POSIT es el algoritmo que primero entra en dicho estado, seguido de RMS y RANSAC.

- *Objeto perdido por variación brusca de pose.* Situación ilustrada en el sector D de la figura 7.30. A medida que el objeto se aleja, el número de outliers aumenta y la convergencia del algoritmo empeora. Respecto al caso anterior de pérdida por interferencia, el tiempo que transcurre desde que POSIT y RMS dejan de converger hasta que lo hace RANSAC es superior. Esto es debido a que el objeto sigue visible, a diferencia del caso anterior. El algoritmo que mejor se adapta a dicha transformación de pose es RANSAC. En la figura 7.32 se presenta la secuencia.

- *Robustez de SURF.* En las figuras 7.24 y 7.25 podemos comprobar visualmente como, con independencia del contexto o la pose, los puntos característicos extraídos sobre la estructura de los objetos satélite y BACI son idénticos para la mayoría de las tomas. De esta manera, tenemos una base de puntos robusta para establecer correspondencias válidas y converger a la solución deseada.

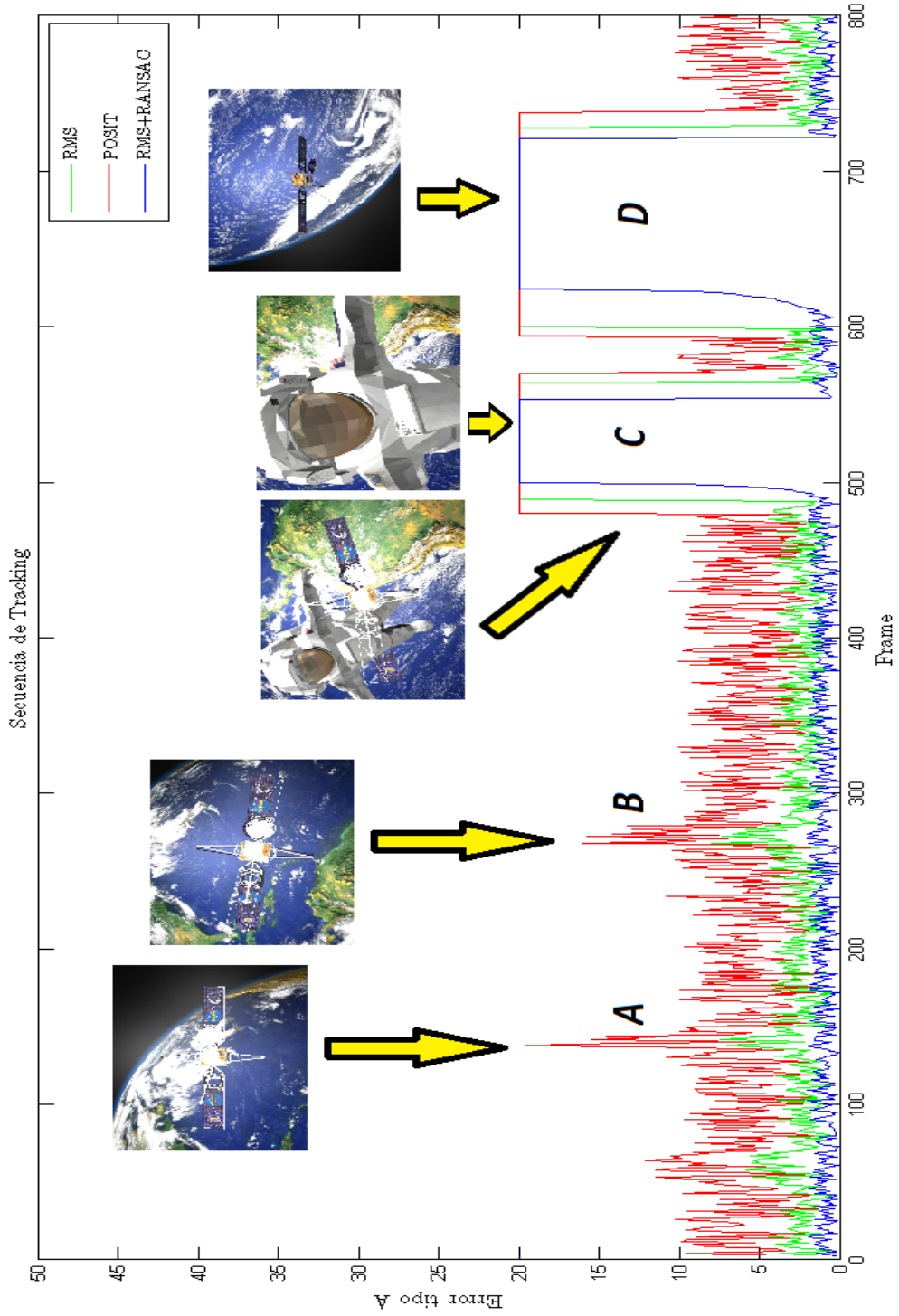


Figura 7.30: Error numérico sobre secuencia tracking satélite.

Objeto Perdido: Interferencia

Estimación Resultado	Correspondencias	Puntos característicos

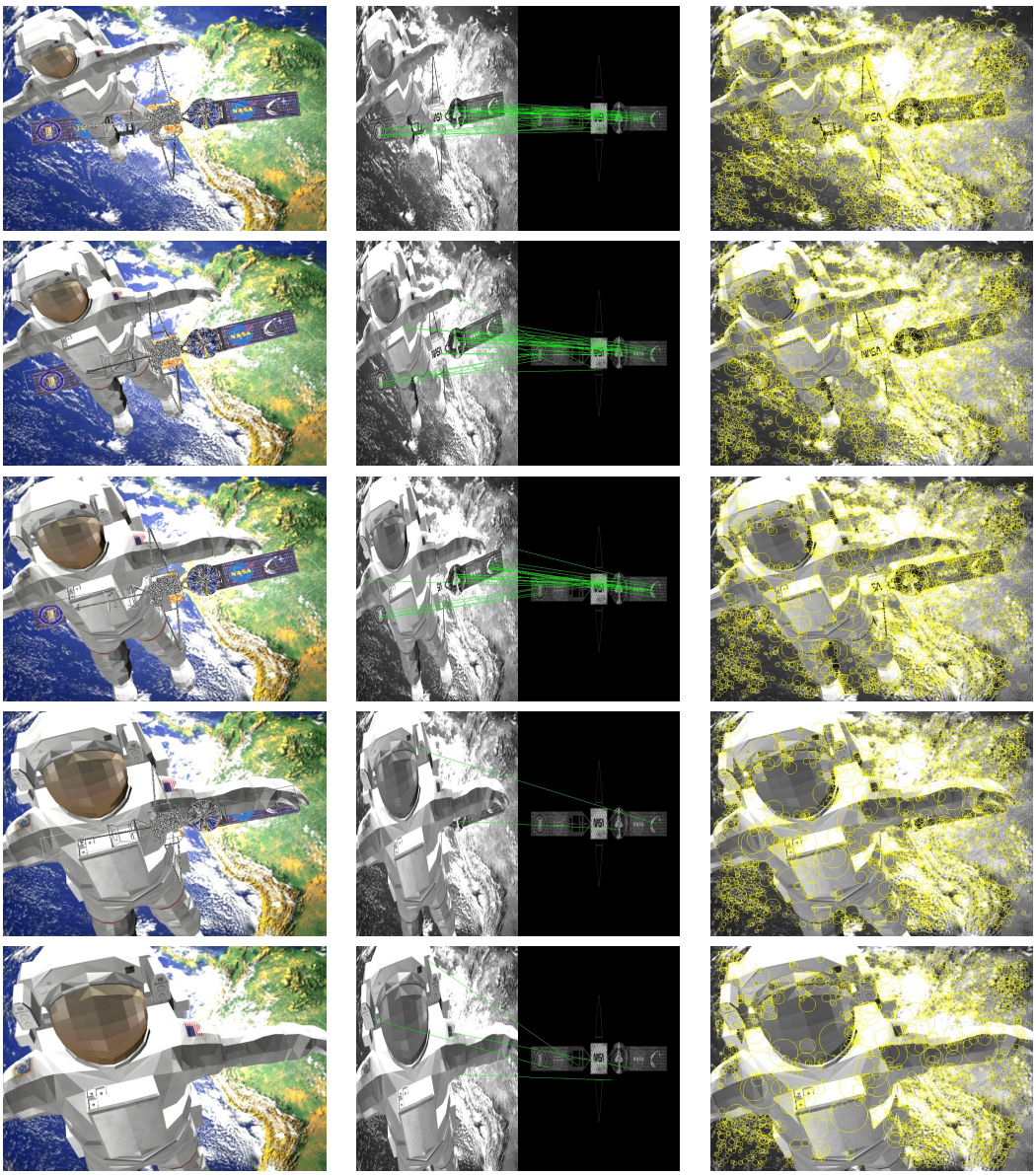


Figura 7.31: Interferencia objeto en imagen.

Objeto Perdido: Pose alejada

Estimación Resultado	Correspondencias	Puntos característicos

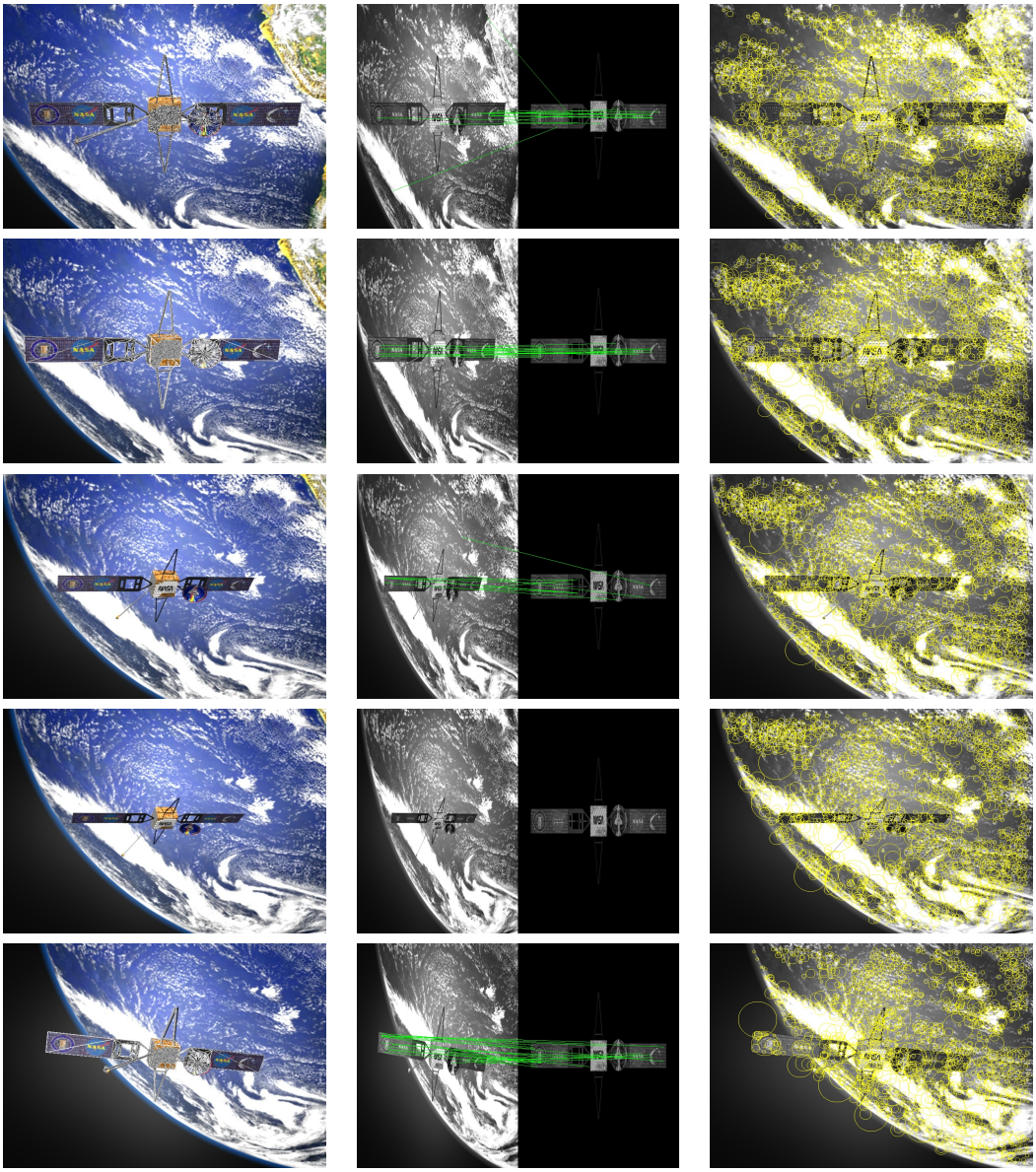


Figura 7.32: Objeto perdido, pose alejada.

Parte IV

Conclusiones y trabajos futuros

Capítulo 8

Conclusiones

El proyecto aborda el problema de estimación de pose 3D. Como se ha visto, existen múltiples alternativas que difieren en la filosofía a la hora de plantear el problema. Entre ellas, se opta por la utilización de información visual basada en puntos característicos sobre la imagen, presentándose tres soluciones. La primera de ellas es POSIT, algoritmo ampliamente utilizado en el campo de la visión por ordenador. Sin embargo, presenta resultados poco robustos ante outliers, lo que explica la implementación del algoritmo propuesto basado en mínimos cuadrados. En este sentido se desarrollan dos alternativas que difieren en el tratamiento de los datos corruptos. La primera de ellas utiliza una etapa de eliminación 3σ y la segunda implementa RANSAC. El objetivo es la evaluación de las prestaciones de dichos algoritmos para determinar el contexto y aplicación de cada uno de ellos.

8.1. Contribuciones del proyecto

Las principales contribuciones que se han realizado son:

- Realizar un *estudio del arte* de los diferentes métodos y algoritmos existentes para solucionar el problema de estimación de pose 3D. Además, se realiza un estudio comparativo sobre las técnicas de extracción de puntos de interés más relevantes, tales como los algorit-

mos de Harris, Harris-Laplace, SUSAN, SIFT y SURF. Por último, se presentan los resultados de ajuste de los distintos parámetros que intervienen en las fases de establecimiento de correspondencias y evaluación de descriptores.

- *Evaluación de algoritmos de estimación de pose 3D.* Se presenta un estudio completo de comparación de los diferentes métodos implementados en múltiples escenarios, analizándose los parámetros y modificaciones oportunas a realizar en función del contexto. De esta manera, se establece un punto de referencia para el desarrollo de nuevos proyectos en los que tomando como referencia dichos resultados, se pueda elegir el algoritmo y los parámetros adecuados que optimicen la estimación.

- *Plataforma de uso.* Se ha realizado la implementación de una plataforma que permite integrar, de forma sencilla e independiente, el módulo deseado (algoritmo de extracción de puntos de interés, método de estimación de pose, etapa de matching) en la estructura del algoritmo global. De esta manera, se constituye una plataforma base para la implementación de algoritmos de estimación de pose 3D. Además, se ha desarrollado una *plataforma visual* que permite evaluar, en tiempo real, la contribución de cada una de las etapas al resultado final, lo que permite al usuario realizar cambios de manera interactiva y evaluar sus consecuencias.

8.2. Conclusiones más significativas

El estudio que se propone es la evaluación de algoritmos de estimación de pose 3D. En este sentido, el proyecto ofrece resultados globales, que se refieren a las prestaciones del algoritmo como tal y resultados locales relativos a cada una de las etapas que lo componen.

Conclusiones de etapas intermedias

Los algoritmos de *extracción de puntos de interés* SURF y SIFT presentan resultados robustos, superando en gran medida las prestaciones aportadas por SUSAN, Harris o Harris-Laplace. Las secuencias de tracking avalan la calidad del descriptor SURF, produciendo resultados aceptables en contextos extremos de ruido, escenario y transformación geométrica. Las implementaciones actuales de este tipo de detectores ofrecen información local adicional que ha de ser utilizada para mejorar las prestaciones del algoritmo.

La etapa de *filtrado inicial*, basada en la información adicional de los descriptores y en la localización espacial del punto de interés, mejora las prestaciones en términos de coste y tiempo computacional del algoritmo. Además, en el contexto de refinamiento de pose, en la que se parte de una estimación cercana a la real, mejora de manera significativa la convergencia. Para caracterizar su importancia, se procedió al análisis de dicha etapa ante contextos de visualización parcial, cambio de escenario y ruido en el proceso de formación de la imagen. En todos ellos resultó ser de gran ayuda y fundamental en la aplicación de algoritmos de extracción de puntos de interés. Si la pose del objeto está alejada de la pose inicial, es necesario desactivar dicha etapa para realizar búsquedas globales sobre la imagen. Sin embargo, como queda demostrado en el informe, una vez estimada la nueva pose, es necesario volver a activarla para refinar los resultados de estimación.

Respecto a la filosofía de *elección de parámetros de la fase de correspondencias*, se planteaban dos alternativas: Uso de parámetros restrictivos, en este sentido se producían asociaciones correctas, si bien el número de correspondencias es reducido. La segunda alternativa es tomar parámetros más flexibles, incrementándose el número de asociaciones erróneas y utilizar en paralelo algoritmos de tratamiento de datos corruptos. Como queda demostrado en el informe, la segunda alternativa es preferible y en la mayoría de los casos necesaria.

Conclusiones globales

Los resultados de tracking, así como el estudio de las transformaciones específicas presentadas permiten obtener conclusiones respecto al rango, contexto y escenario de aplicación para cada uno de los algoritmos. Existen limitaciones físicas generales, casos en los que las etapas de extracción de características son incapaces de ofrecer la información necesaria. Sin embargo, el rango de funcionamiento y precisión depende de cada algoritmo en particular, produciéndose resultados diferentes en escenarios idénticos.

Las pruebas realizadas caracterizan los algoritmos de estimación de pose 3D ante transformaciones geométricas de rotación y translación, así como escenarios de ruido variable. La secuencia de tracking estudiada ofrece resultados interesantes sobre el *error medio* y *varianza* de la estimación. Ésta información es clave para determinar el *rango de funcionamiento* de cada algoritmo en particular, y en contextos tales como el alejamiento del objeto, pérdida parcial de información visual por interferencia o cambio brusco de pose, permite detectar rápidamente el estado en el que se encuentra el proceso y actuar en consecuencia.

POSIT es el algoritmo que peor se ajusta ante cambios en la imagen real. Además, es un método que depende en gran medida de la calidad o exactitud de sus datos de entrada, conjunto de puntos característicos asociados en la imagen real y modelo. En este sentido, pequeñas variaciones en dicho conjunto producen una elevada varianza en los resultados.

La optimización vía mínimos cuadrados mejora las prestaciones de convergencia de POSIT, aumentando su rango de funcionamiento. La varianza en la estimación es reducida, produciéndose resultados robustos y constantes ante escenarios similares. La incorporación de RANSAC optimiza los resultados anteriores produciendo valores mínimos de varianza y error medio.

8.3. Perspectivas y trabajo futuro

A lo largo del estudio y desarrollo del proyecto surgen ideas de mejora y optimización del algoritmo. Las principales líneas de investigación de este proyecto son:

- *Objetos con poca Textura.* Si el objeto bajo estudio tiene poca textura, algoritmos como SURF o SIFT extraerían poca información de la imagen. En este contexto, tendría más sentido la utilización de otro tipo de algoritmos de extracción de características. En la bibliografía se proponen otras alternativas que identifican la forma geométrica del objeto y que en este contexto pueden ser de utilidad. Por supuesto, las alternativas basadas en puntos de interés y forma geométrica pueden combinarse y aplicarse de forma conjunta. Es decir, si sabemos que el objeto tiene textura en cierta región, podemos aplicar algoritmos como SURF o SIFT de manera local y en el resto de la estructura aplicar algoritmos que identifiquen formas geométricas.

- *Etapa de extracción de puntos de interés.* Los algoritmos SIFT y SURF utilizados son el punto de referencia hacia nuevos métodos más robustos y flexibles. Existen múltiples mejoras, si bien el tiempo computacional es elevado ya que realizan varias iteraciones sobre el algoritmo base.
  Los algoritmos propuestos utilizan imagenes en escala de grises. En este sentido, la información de color puede ser utilizada para compensar, complementar o reemplazar a un descriptor de forma geométrica, dado el volumen de información que se puede extraer del modelo de color [32].

- *Etapa de tratamiento de datos corruptos.* Ha quedado demostrado que RANSAC es el algoritmo que mejor se adapta a contextos en los que se producen altos porcentajes de outliers en la muestra. Sin embargo, existen múltiples modificaciones sobre el algoritmo base que optimizan las prestaciones de convergencia, tiempo y coste

computacional del algoritmo global. [28] estudia las mejoras más populares de RANSAC.

- *Elección de parámetros de forma dinámica.* El estudio realizado propone el ajuste de los parámetros asociados a cada etapa del algoritmo en función a la tipología del objeto o contexto. En este sentido, es interesante disponer de un método que fijase dichos valores de manera automática, presentando resultados óptimos con independencia del escenario y sin necesidad de ajuste previo.

- *Tracking.* El objetivo principal del proyecto es el desarrollo y evaluación de algoritmos de estimación de pose 3D de un objeto. La fase de seguimiento ha tenido poca importancia y tan sólo se ha realizado una pequeña aproximación. El uso de filtros temporales tipo Kalman suavizaría la respuesta de estimación y minimizaría el error de predicción. En este sentido, existe un amplio margen de mejora.

Parte V

Anexos

Apéndice A

Presupuesto

En este anexo se presenta el presupuesto global del proyecto. En términos generales, el presupuesto se desglosa en honorarios y equipos.

A.1. Equipos

En la tabla A.1 se desglosa el conjunto de equipos necesarios para la realización del proyecto.

El software utilizado es linux en su versión Kubuntu y por lo tanto es gratuito. Además se utilizarán programas como Blender (modelado de entornos 3D), qtCreator (plataforma de programación en C), meshLab (visor y editor de modelos 3D) o Gimp (editor fotográfico) que son también gratuitos.

Concepto	Precio (Euros)
Portátil Toshiba Satellite A665	860
Logitech Webcam Pro 9000	80
Cámara PMD CamCube 2.0 (ver anexo C)	8000
TOTAL	8940

Cuadro A.1: Desglose de tareas y tiempo utilizado.

Tarea	Horas
Documentación	60
Estudio y adaptación a Blender, qtCreator y meshlab	40
Implementación del algoritmo RMS	230
Pruebas del algoritmo de estimación de pose 3D RMS	80
Implementación POSIT, RANSAC	40
Pruebas comparación algoritmos	40
Redactar memoria	100
TOTAL	559

Cuadro A.2: Desglose de tareas y tiempo utilizado.

Responsabilidad	Precio (Euros)
Ingeniero desarrollador	14794
Tutor, dirección	1035.6
IVA (18 %)	2532.8
TOTAL	18363

Cuadro A.3: Honorarios.

A.2. Honorarios.

Para la realización de los honorarios se toma como referencia los baremos del COIT (Colegio Oficial de Ingenieros de Telecomunicación). En la tabla A.2 se desglosa el conjunto de tareas realizadas durante el tiempo que dura el proyecto, ocho meses.

Teniendo en cuenta el sueldo medio de un ingeniero, 26 euros a la hora, se produce un total de 14794 euros tan sólo considerando la contribución del ingeniero al cargo. Además, hay que añadir el tiempo estimado dedicado por el tutor en su trabajo de dirección del proyecto, entorno a un 7 %. En la tabla A.3 se presenta el resultado total de los honorarios.

Concepto	Precio (Euros)
Equipos	8940
Honorarios	18363
TOTAL	27303

Cuadro A.4: Presupuesto total.

A.3. Presupuesto final

En la tabla A.4 se presenta el resultado global del presupuesto, considerando contribución de honorarios y equipos. El presupuesto total asciende a veintisiete mil trescientos tres euros.

Apéndice B

Algoritmo POSIT

En este anexo se presentan las relaciones matemáticas que vertebran el algoritmo de estimación de pose $3D$ POSIT.

La matriz de rotación se define como:

$$R = \begin{bmatrix} i_u & i_v & i_w \\ j_u & j_v & j_w \\ k_u & k_v & k_w \end{bmatrix}$$

donde i y j se calculan a partir de las coordenadas en la imagen y k se calcula como el producto vectorial entre ambas.

Tomando como referencia la figura 4.8, si Z_0 (profundidad del punto de referencia M_0) se conoce, entonces podríamos determinar el resto de coordenadas X_0, Y_0 de M_0, ya que el vector de translación $T = OM_0$ se alinea con el vector Om_0 como se puede ver en la figura 4.8. Estos vectores se pueden expresar como $T = \frac{Z_0}{f} Om_0$ donde f es la distancia focal conocida.

En relación a los puntos m_x, se establecen las siguientes relaciones:

$$x_i = \frac{fX_i}{Z_i}$$

$$y_i = \frac{fY_i}{Z_i}$$

Por otro lado, considerando la proyección que se realiza a partir de los puntos p_x, obtenemos:

$$x'_i = \frac{fX_i}{Z_0}$$

$$y'_i = \frac{fY_i}{Z_0}$$

Combinando ambas ecuaciones:

$$x'_i = \frac{f}{Z_0}X_0 + \frac{f}{Z_0}(X_i - X_0) = x_0 + s(X_i - X_0)$$

$$y'_i = y_0 + s(Y_i - Y_0)$$

donde $s = \frac{f}{Z_0}$ se conoce como el factor de escala.

El siguiente paso relaciona las variables no conocidas i,j yZ_0(profundidad del punto M_0), con las variables conocidas M_0M_i, vectores del objeto que contiene las coordenadas x_i e y_i de los puntos m_i. Se presentan las siguientes relaciones:

$$M_0M_i\frac{f}{Z_0}i = x_i(1 + \epsilon_i) - x_0$$

$$M_0M_i\frac{f}{Z_0}j = y_i(1 + \epsilon_i) - y_0$$

donde $\epsilon_i = \frac{1}{Z_0}M_0M_ik$ y $k = i \times j$ como ya quedó definido. El parámetro ϵ se inicializa a 0, por lo tanto podemos considerar que es un parámetro dado. Gracias a las ecuaciones anteriores, podemos expresar el siguiente sistema de ecuaciones lineales:

$$M_0M_iI = \tau_i$$

$$M_0 M_i J = \lambda_i$$

donde

$$I = \frac{f}{Z_0} i$$

$$J = \frac{f}{Z_0} j$$

$$\tau_i = x_i \left(1 + \epsilon_i\right) - x_0$$

$$\lambda_i = y_i \left(1 + \epsilon_i\right) - y_0$$

Por último, si tenemos en cuenta el conjunto completo de correspondencias entre la imagen real y modelo, obtenemos el siguiente par de ecuaciones lineales:

$$AI = x'$$

$$AJ = y'$$

donde x' e y' son los vectores compuestos por los términos τ_i y λ_i respectivamente.

Los parámetros I y J se calculan mediante una descomposición en valores singulares o autovalores. Una vez calculados dichos valores, se determinan los parámetros i y j mediante un proceso de normalización. Además, tomando la norma de los vectores I ,J y aplicando las relaciones implícitas del factor de escala, se obtiene Z_0. La pose $3D$ estimada del objeto se reconstruye de manera directa a partir de este conjunto de parámetros.

Apéndice C

Levenberg-Marquardt

El algoritmo de Levenberg-Marquardt (LMA) proporciona una solución al problema de optimización de una función, generalmente no lineal, sobre un espacio de parámetros asociados a la misma. En realidad, es una mezcla de dos algoritmos, el método de *Gauss-Newton* y *descenso de máxima pendiente*. En términos de robustez, LMA supera las prestaciones del algoritmo Gauss-Newton, lo que significa, que en la mayoría de los casos converge a la solución correcta, aun partiendo de una posición inicial alejada. Sin embargo, para cierto conjunto de funciones lineales y valores iniciales, el tiempo de convergencia que presenta LMA es superior. En general, el algoritmo LMA se considera como una mejora del algoritmo de Gauss-Newton.

La aplicación fundamental del algoritmo Levenberg-Marquardt es el ajuste de curvas a partir de mínimos cuadrados. Dado un conjunto de pares de datos empíricos m, de las variables dependientes e independientes (x_i, y_i), se pretende optimizar los parámetros β de la curva modelo $f(x, \beta)$, tal que la suma de las desviaciones cuadráticas sea mínima, es decir:

$$S(\beta) = \sum_{i=1}^{m} [y_i - f(x_i, \beta)]^2$$

Al igual que otros algoritmos de optimización, el algoritmo de Levenberg-Marquardt es un algoritmo iterativo. El proceso comienza cuando se in-

troduce una estimación, o inicialización del parámetro β. En la mayoría de los casos, suele funcionar bien una aproximación uniforme del tipo: $\beta^T = (1, 1, ..., 1)$. En otros casos, el algoritmo sólo converge, si la estimación inicial está cerca del resultado final.

En cada iteración, el parámetro β se actualiza. De esta manera $\beta_i = \beta_{i-1} + \delta$. El cálculo del parámetro δ se realiza mediante una linealización de las funciones $f(x_i, \beta + \delta)$ como sigue:

$$f(x_i, \beta + \delta) \approx f(x_i, \beta) + J_i\delta$$

donde $J_i = \frac{\partial f(x_i,\beta)}{\partial \beta}$ es el gradiente de f con respecto a β.

En el mínimo de dicha función, el gradiente de la función $S(\beta)$ con respecto al parámetro δ es nulo. Por otro lado, la aproximación de primer orden de la función $f(x_i, \beta + \delta)$ se expresa como:

$$S(\beta + \delta) \approx \sum (y_i - f(x_i, \beta) - J\delta)_{i=1}^{m}$$

en notación vectorial:

$$S(\beta + \delta) \approx \|y - f(\beta) - J\delta\|^2$$

Tomando derivadas con respecto a δ y considerando el resultado nulo, se obtiene:

$$\left(J^T J\right)\delta = J^T [y - f(\beta)]$$

donde J representa la matriz jacobiana cuya fila i-ésima se representa como J_i. Por otro lado, los vectores f e y representan la componente i-ésima de $f(x_i, \beta)$ e y_i respectivamente. De esta manera, se establecen un conjunto de ecuaciones lineales, cuya resolución da como resultado el valor de δ.

La contribución de Levenberg fue la sustitución de la ecuación anterior por una versión amortiguada:

$$\left(J^T J + \lambda I\right)\delta = J^T [y - f(\beta)]$$

donde I representa la matriz identidad como término adicional al vector estimado β. Por otro lado, el factor de amoritigüamiento λ, se ajusta en cada iteración y está en directa relación con la velocidad de convergencia. Teniendo en cuenta que el gradiente de la función S con respeto a β sigue la expresión $-2\left(J^T\left[y - f\left(\beta\right)\right]\right)$, entonces, si se determinan valores altos de λ, los pasos se toman en la dirección del gradiente.

El algoritmo termina si se produce una de las siguientes razones.

- El valor del parámetro δ cae por debajo de cierto límite establecido.
- El valor resultado $\beta + \delta$ es aceptable, es decir, está dentro de los límites deseados, establecidos al comienzo del algoritmo.

El parámetro β estimado en la última iteración se toma como resultado final del algoritmo.

Apéndice D

Cámara PMD CamCube 2.0

En este anexo se realiza la presentación de la cámara TOF utilizada en el desarrollo del proyecto, PMD CamCube 2.0.

CamCube 2.0 es la cámara Time of Flight con resolución más alta en la actualidad. El sensor óptico es de 204×204 pixels, permitiendo capturas en tiempo real con información de escala de gises y profundidad. Gracias a su elevada sensibilidad y a la incorporación de nuevos módulos PhotonICs PMD 41k-S2, se consiguen velocidades de captura más elevadas y barrer un margen más amplio de distancias. Por otro lado, la integración de un módulo SBI (Suppression of Background Illumination), la consolida como una tecnología sólida para utilizar en ambientes interiores, así como exteriores.

El contexto aplicativo de este tipo de cámaras es amplio. Entre sus principales ámbitos, destacan:

- Robots móviles.
- Industria automovilística.
- Industria aeronáutica.
- Tecnología médica.
- Seguridad.
- Entretenimiento, ocio, electrónica de consumo.

Figura D.1: Cámara PMD CamCube 2.0.

(a) (b)

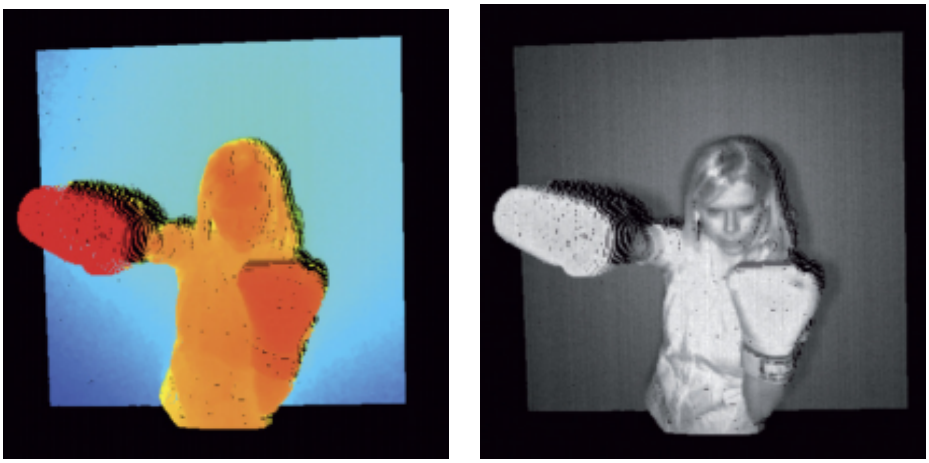
Figura D.2: Módulos PMD CamCube 2.0. En la figura (a) se ilustra el módulo de la cámara y en (b) el módulo de iluminación.

A continuación se detallan sus principales características técnicas:

- Ángulo de visión elevado, optimizado mediante PMD[vision]® optics.
- Resistencia a Motion Blur.
- Flexibilidad en el tipo de medición, con posibilidad de utilizar distintas fuentes de modulación de la fuente de luz.
- Posibilidad de establecer regiones de interés (ROI).
- Posibilidad de utilizar varios canales de frecuencia en función del modo de operación de la cámara.
- Paquete software incorporado para procesado de datos y visualización.
- API e interfaz MATLAB para Linux y Windows (32/64 bits).

D.1. Datos generados

En esta sección se presentan los distintos datos que genera PMD [vision] ® CamCube 2.0. Entre ellos destaca la imagen de distancia, que



(a) (b)

Figura D.3: Datos generados PMD CamCube 2.0. En la figura (a) se presenta la captura 3D con información de color. En (b) se ilustra la imagen 3D con información de escala de grises.

proporciona, en cada pixel, una estimación de profundidad. Si el campo de visión y los parámetros de la cámara se conocen, es posible calcular las coordenadas cartesianas asociadas al modelo.

Además del mapa de distancias, la cámara provee de una imagen de amplitud, que proporciona información adicional útil para determinar la calidad de la estimación. En este sentido, cuanto mayor sea el valor de amplitud asociado a un pixel, más fiable es la distancia que se corresponde con dicho valor. Si la cámara observa la escena con altos valores de reflectividad (en el espectro infrarrojo), los valores de amplitud serán altos y por lo tanto, la estimación será fiable. Los objetos con reflectividad baja producirán estimaciones de distancias ruidosas y por lo tanto bajas amplitudes. En otras palabras, es posible evaluar la calidad del estimador de distancia asociado a un pixel, mirando su valor de amplitud. De esta manera, podemos elegir ignorar distancias por debajo de un cierto umbral, con el objetivo de garantizar que se trabaja con resultados fiables.

Por último, la cámara genera una imagen a color y otra en escala de grises, como se puede comprobar en la figura D.1.

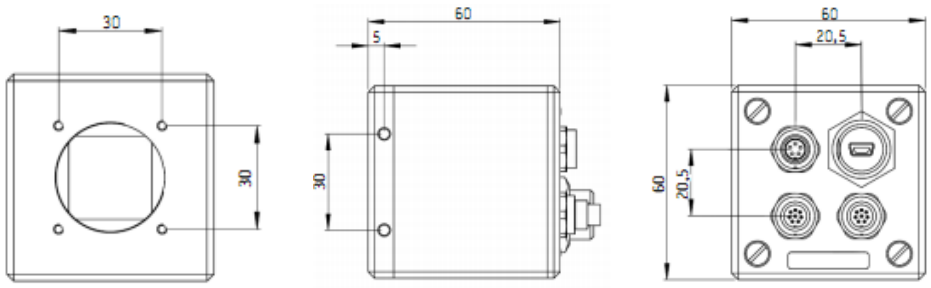
D.2. Parámetros técnicos

A continuación se presenta la tabla de especificaciones:

Parámetro	Valor (Configuración típica)	Notas
Tipo de Sensor	PhotonICs®PMD 41k-S2 (200x200)	Incl. SBI (Suppression of Background Illumination)
Rango de medidas Estándar	0.3 a 7 m	
Repeatability (1σ)	< 3 mm	Typical value, central sensor area @4m distance, 75 % reflectivity
Velocidad de captura (3D)	40 fps @ 200x200 pixels 60 fps @ 176x144 pixels 80 fps @ 160x120 pixels	Typical value, depending on camera settings and ROI
Campo de visión	40° x 40°	CS mount lens: f = 12,8 mm F1,1
Longitud de onda de iluminación	870 nm	Eye safety class 1
Modos de operación	hardware / software trigger mode, free run mode (standard)	
Voltage de alimentación	12V ± 10 %	
Interfaz	USB 2.0	
Temperatura de operación	0°C a 50°C	

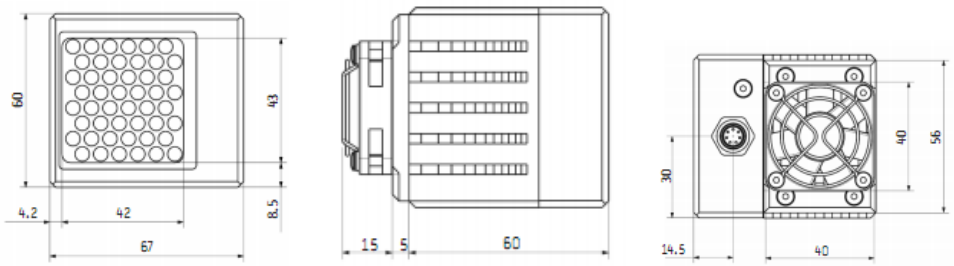
Dimensiones

Cámara



(a) (b) (c)

Módulo de iluminación



(d) (e) (f)

Figura D.4: Dimensiones PMD CamCube 2.0. En las figuras (a), (b) y (c) se ilustran las perspectivas frontal, lateral y trasera de la cámara. En las figuras (c), (d) y (e) se presentan sus correspondientes asociadas al módulo de iluminación. Las dimensiones están expresadas en mm.

Bibliografía

[1] Levitt T.S. and Lawton D.T. *"Qualitative navigation for mobile robots"*. Journal of Artificial Intelligence, Número: 44, Páginas: 305-360, 1990.

[2] Mubarak Shah. Computer Science Deparment, universidad central de Florida. *Fundamentals of Computer Vision*, 1997.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV*, 2008.

[4] R. Hartley. Multilinear relationships between coordinates of corresponding image points and lines. In *Proceedings of the International Workshop on Computer Vision and Applied Geometry*, International Sophus Lie Center, Nordfjordeid, Norway, 1995.

[5] Markus Ulrich, Christian Wiedemann, and Cartesen Steger. *CAD-based Recognition of 3D Objects in Monocular Images,* 2009

[6] Quan and Lan. Linear n-point camera pose determination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1999.

[7] Daniel F. DeMenthon and Larry S. Davis. *Model-based object pose in 25 lines of code.* International Journal of Computer Vision, Páginas: 123–141, 1995.

[8] Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1991.

[9] Yuan. A general photogrammetric solution for the determining object position and orientation. *IEEE Trans. Robotics and Automation*, Páginas: 129–142, 1989.

[10] Dhome. Determination of the attitude of 3d objects from single perspective view. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Páginas: 1265– 1278, 1989.

[11] Lowe. *Perceptual Organization and Visual Recognition,* 1985.

[12] Perwass Rosenhahn and Sommer. *Pose estimation of 3D freeform contours*. Technical Report 0207, University Kiel, 2002.

[13] Nevatia and Ulupinar. *Perceptionof 3D surfaces from 2-d contours*, 1993.

[14] Fischer and Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, Páginas: 381–395, 1981.

[15] David Eugene Smith. History of modern mathematics. *Mathematical Monographs,* 2005.

[16] Cheryl Weber Sklair William J. Wolfe, Donald Mathis and Michael Magee. The perspective view of three points. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1991.

[17] R. Y. Tsai. Versatile camera calibration technique for highaccuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 1987.

[18] C. Harris y M. Stephens: *"A combined corner and edge detector"*. Proceedings of the 4th Alvey Vision Conference, Páginas: 147 - 151, 1988.

[19] S. M. Smith and J.M Brady: *"SUSAN-A new approach to low level image processing"*. International Journal of Computer Vision, Volumen: 23, Número: 34, Páginas: 45-78, 1997.

[20] David G. Lowe: *"Object recognition from local scale-invariant features"*. Seventh IEEE International Conference on Computer Vision, Volumen: 2, Páginas: 1150-1157, 1999.

[21] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool: *"SURF: Speeded Up Robust Features"*. Computer Vision and Image Understanding (CVIU), Volumen: 110, Número: 3, Páginas: 346–359, 2008.

[22] Johannes Bauer, Niko Sünderhauf, Peter Protzel: *"Comparing Several Implementations of two Recently Published Feature Detectors"*. In Proc. of the International Conference on Intelligent and Autonomous Systems (IAV), Septiembre 2007.

[23] Philip David, Daniel DeMenthon, Ramani Duraiswami and Hanan Samet: *"SoftPOSIT: Simultaneous Pose and Correspondence Determination"*. 2002.

[24] S. Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. *A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions*, 2010. URL http://canesta.com.

[25] G. J. Iddan and G. Yahav. 3*D Imaging in the Studio (and elsewhere ...)*. Proc. of SPIE, 2001.

[26] PMDtec. PMD[vision]® CamCube 2.0, 2010. URL http://www.pmdtec.com.

[27] Microsoft. The Kinect Project: Introducing Controller-Free Gaming and Entertainment, 2010. URL http://www.xbox.com/en-US/kinect.

[28] R. Raguram, J.M. Frahm and M. Pollefeys. *A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus*. Computer Vision-ECCV. Páginas: 500-513, 2008.

[29] Z. Zhang. *Flexible camera calibration by viewing a plane from unknown orientations*, in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference. Volumen 1, 1999.

[30] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab*. URL http://www.vision.caltech.edu/bouguetj/calib_doc/

[31] N. Burrus, J. García, L. Moreno and M. Abderrahim. *3D Object Model Acquisition and Recognition with a Time-of-Flight camera*. In 7th Robocity2030 Workshop on Computer Vision, 2010.

[32] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. *Evaluation of color descriptors for object and scene recognition*. In Pattern Analysis and Machine Intelligence. Páginas 1-8, 2009.

[33] Nicolas Burrus, Thierry M. Bernard and Jean-Michel Jolion. *Bottom-Up and Top-Down Object Matching Using Asynchronous Agents and a-Contrario Principles*. International Conference on Vision Systems. Volumen: 5008, Páginas: 343-352, 2008.